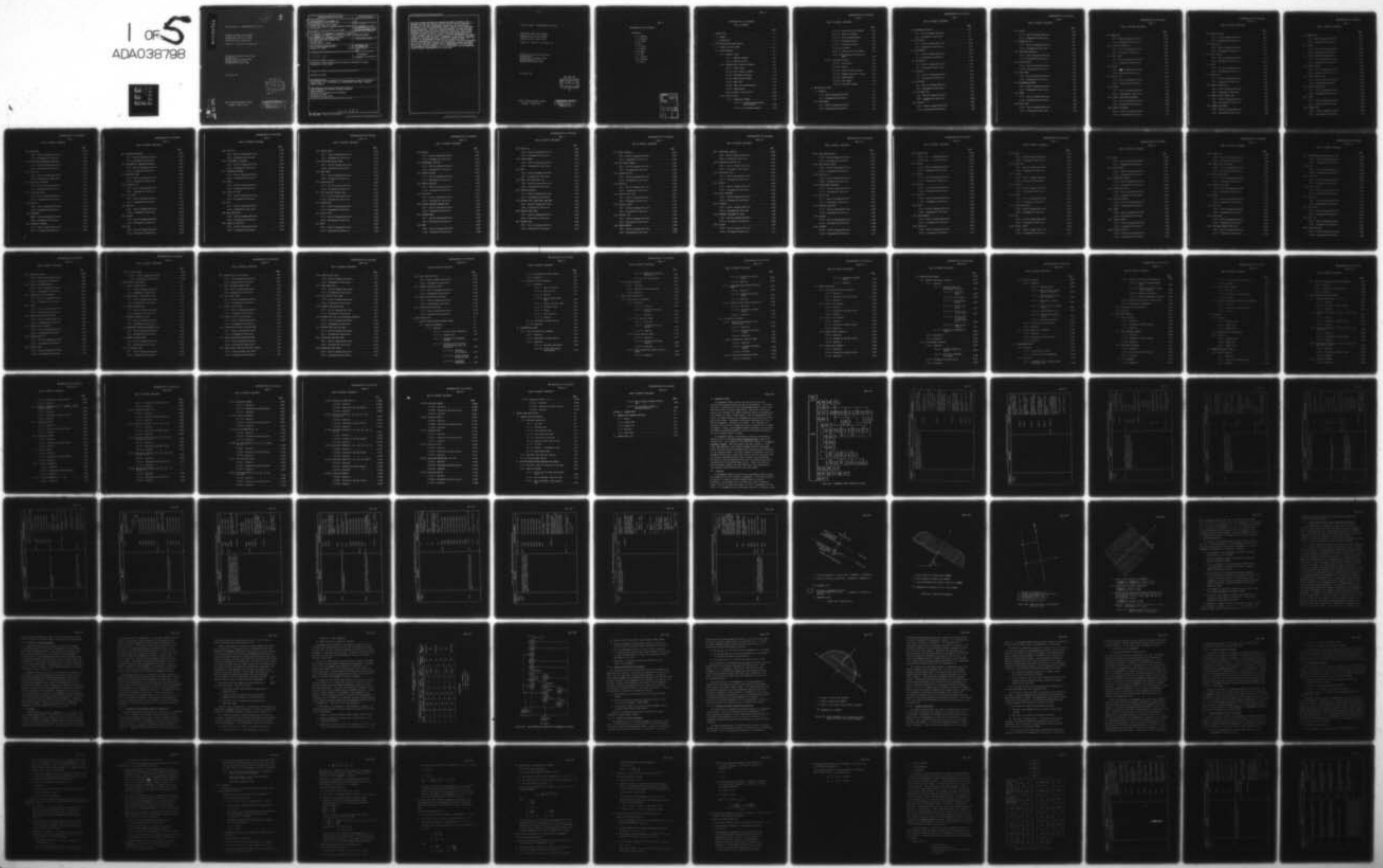
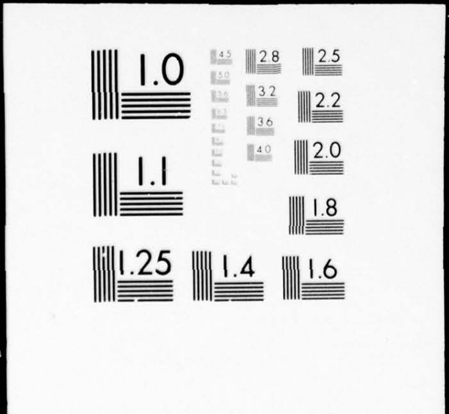
AD-A038 798 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

1 OF 5
ADA038798





NAVTRAEQUIPCEN 73-C-0156-E003

2 4

B.S.

Technical Report: NAVTRAEQUIPCEN 73-C-0156

MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAINING SIMULATOR (CATTS), DEVICE 16A3

Volume III - Sections 5.6 through 5.10

TRW Defense and Space Systems Group
One Space Park
Redondo Beach, California 90278

Contract N61339-73-C-0156
NAVTRAEQUIPCEN Task No. 3853

28 January 1977

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 19 NAVTRAEQUIPCEN 73-C-0156-E003-3 / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 Mathematical Model User's Manual Combined Arms Tactical Training Simulator (CATTS), Device 16A3. Volume III. Sections 5.6 through 5.10.	5. TYPE OF REPORT & PERIOD COVERED Final Report. - 28 Jan 77
	6. PERFORMING ORG. REPORT NUMBER NAVTRAEQUIPCEN Task #3853
7. AUTHOR(s) 10 D.S./Adamson, E.C./Andreani, G.W./Archer, R./Cho; J.H./Jensen; H.R. Milder; D.W. Myers; C.R. Peer; D.T. Unangst; W.H. Williams; J.A. Wulfe	8. CONTRACT OR GRANT NUMBER(s) 15 N61339-73-C-0156
9. PERFORMING ORGANIZATION NAME AND ADDRESS TRW Defense and Space Systems Group One Space Park Redondo Beach, CA 90278	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Orlando, FL 32813	12. REPORT DATE 11 28 January 1977
	13. NUMBER OF PAGES 1846
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 456p.	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Introduction, System Organization and Overview, Definition of Terms, Description of Data Tables

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for Public

18. SUPPLEMENTARY NOTES

This Technical Report prepared by TRW Corp (Thompson, Ramos and Williams) of Redondo Beach, CA., as specified by NAVTRAEQUIPCEN Task #3853, Contract N61339-73-C-0156

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CATTS (Combined Arms Tactical Training Simulator)
Systems Overview
CATTS Software (Background and Foreground)
Definition of Terms
Description of Data Tables

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

409 637

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

LB

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This user's manual describes the mathematical model contained in the Combined Arms Tactical Training Simulator (CATTS) from the viewpoint of providing the user with the necessary insight into the structure and performance of the CATTS mathematical model in support of its operational use and maintenance. CATTS is a computer-supported, two-sided training simulator that is used to provide effective training for battalion field commanders and staff officers by realistically simulating ground combat operations between red and blue forces. The CATTS mathematical model is a large, detailed computer time-step simulation of the tactical battlefield environment, including detections, engagements, weapon firings, casualties, movement, and environmental effects for up to ninety-nine units. A complete understanding of the CATTS system operation, structure, performance and the use of the CATTS software can be constituted by using this manual together with the CATTS Operators Manual (NAVTRAEQUIPCEN 73-C-0156-E001), the Programming Report (NAVTRAEQUIPCEN 73-C-0156-A008), and the superindex program listings (NAVTRAEQUIPCEN 73-C-0156-A008).

Technical Report: NAVTRAEQUIPCEN 73-C-0156

MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAINING SIMULATOR (CATTS), DEVICE 16A3

Volume III - Sections 5.6 through 5.10

TRW Defense and Space Systems Group
One Space Park
Redondo Beach, California 90278

Contract N61339-73-C-0156
NAVTRAEQUIPCEN Task No. 3853

28 January 1977

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

NAVTRAEQUIPCEN 73-C-0156-E003

Prepared by:

D. S. Adamson

E. C. Andreani

G. W. Archer

R. Cho

J. H. Jensen

H. R. Milder

D. W. Myers

C. R. Peer

D. T. Unangst

W. H. Williams

V. A. Wulff

NAVTRAEQUIPCEN 73-C-0156-E003

TABLE OF CONTENTS

NAVTRAEQUIPCEN 73-C-0156-E003

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

5.6 ENGAGEMENTS MODULE

The Engagements Module contains the logic which governs the positioning of opposing ground units during confrontations. This module functions in conjunction with the Ground Movement Module to direct units toward the center of greatest enemy threat. Specifically, during the course of unit movement, this module determines whether an engagement can be initiated as a result of enemy contact. Once an engagement has been formed, this module maintains the engagement by mobilizing opposing units against one another. Furthermore, the eventual withdrawal by the unit from the engagement is handled by this module. This withdrawal will ultimately cause the unit to change its state of operation and its manner of movement. Finally, the Engagements Module coordinates with the Ground Fire Module by determining whether a unit will be allowed to discharge direct-fire weapons against its enemy.

The Engagements Module is described in a detailed, subroutine-by-subroutine level in the CATTS Trainer Programming Report, pages 5-371 through 5-412. Each subroutine is also described in detail in the MAFIA V Programmer's Manual. Very few changes, other than increases in table sizes, have been made to this module throughout the CATTS development period, and these two earlier documents still provide accurate information for the interested reader. This User's Manual will therefore discuss the Engagements Module from a higher-level, overall point of view rather than an individual subroutine point of view, in order to avoid unnecessary duplication of previous documentation as well as provide the overview of the module which has heretofore been missing. Figure 5-93 shows the module subroutine linkage, and Table 5-45 gives a brief description of each subroutine along with principal inputs and outputs.

5.6.1 Overview

The Engagements Module along with the Ground Movement Module controls the movement of all units involved in engagements. During the simulation, a ground unit is moved in discrete segments toward its desired destination. Each movement step is followed by a check to determine whether the unit has come close enough to initiate an engagement with the enemy. If an engagement can be established, the unit is instructed to move

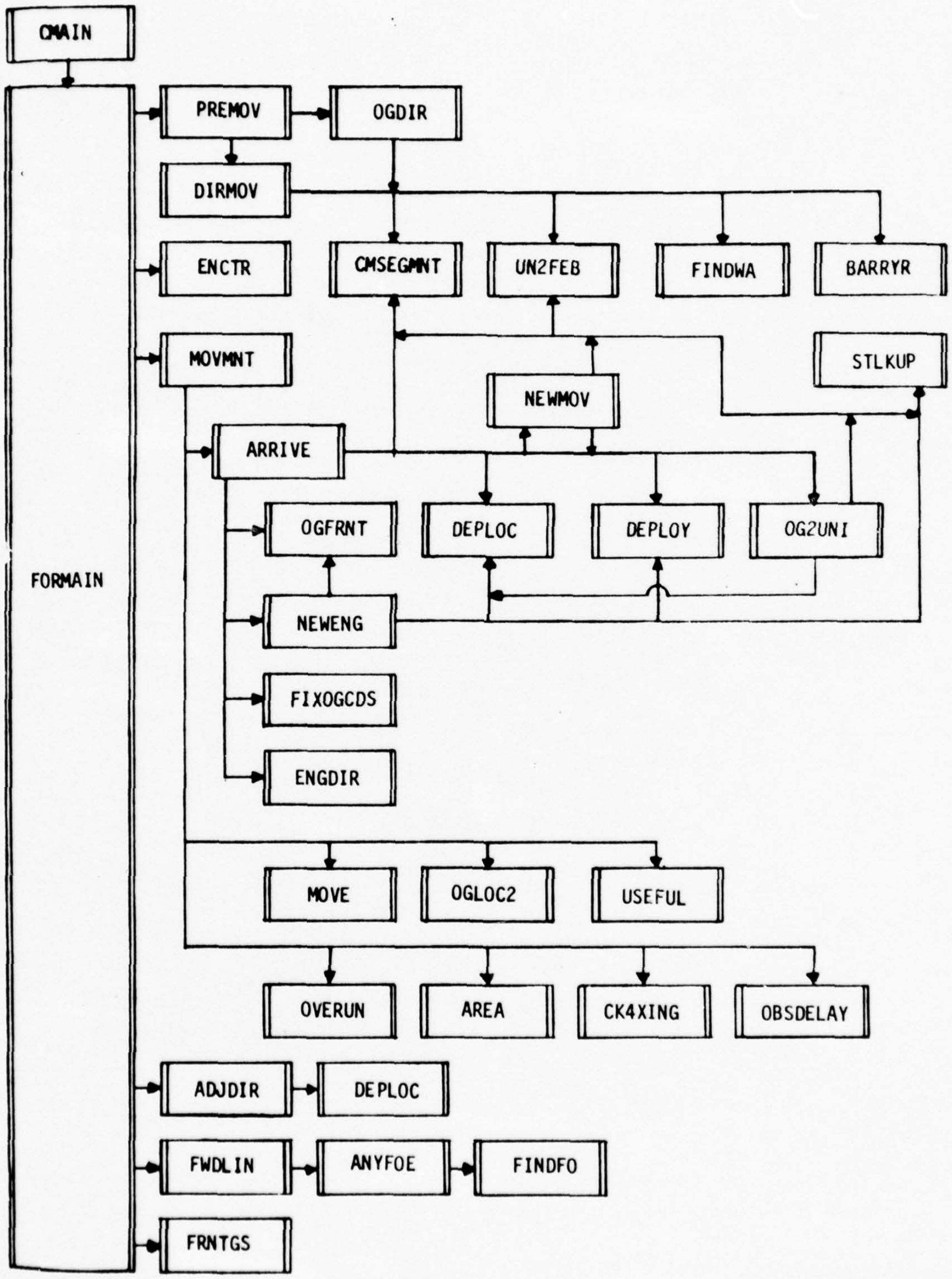


Figure 5-93. Engagements Module Subroutine Linkage

Table 5-45. Engagements Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs		
ADJDIR	Called once per time-step to determine one-sided engagements, calculate the new engagement axis for each, and redeploy each unit and op group in the engagement to correspond with the new axis. See Figures 5-94, 5-95.	Input:	IBFRNT(K)	- Half-frontage of the blue forces in the K-th engagement (meters). (See Figure 5-95.)
			IDPCOD(J)	- Deployment code for J-th operational grouping. 0 = not deployed, cannot move 1 = not deployed, can move 2 = technically deployed 3 = actually deployed

			IENDP1(L)	- Depth of region in which close support targets may be assigned (meters). Measured either from friendly or enemy FEBA, depending on fire support weapon type.
				L = 1 For red forces
				L = 2 For blue forces

			IFEBAB(K,I)	- X coordinate (I = 1) and Y coordinate (I = 2) of the center of the blue FEBA in the K-th engagement. (See Figure 5-95.)
			IFEBAR(K,I)	- Same as IFEBAB, except for red forces.
			IRFRNT(K)	- Same as IBFRNT, except for red forces.
			IXPHIB(K)	- Position of blue FEBA center in direction of engagement

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
ADJDIR (Cont'd)		axis (as defined by DIREAX) for K-th engagement. Set to -9999999 if engagement does not exist.
		IXPHIR(K) - Same as IXPHIB, except for red forces.
		MTCDOG(J) - Movement code of J-th operational grouping. Corresponds to unit variable MVTCD.
		MVDTA1(J) - Movement data value 1 for J-th operational grouping. Corresponds to unit variable MVDT1.
		MVDT1(IU) - Movement data value 1 for unit IU. See Section 5.4 for a full explanation.
		MVTCD(IU) - Movement code of unit IU. See Section 5.4 for a full explanation.
		Output: DIREAX(K,I) - Sine (I = 1) and cosine (I = 2) of engagement axis direction for K-th engagement. Always calculated from blue to red, that is, from (IFEBAB(K,1), IFEBAB(K,2)) to (IFEBAR(K,1), IFEBAR(K,2))

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ADJDIR (Cont'd)			DPMVT(J,I)	- Sine (I = 1) and cosine (I = 2) of direction faced by J-th operational grouping.
			FDIR(IU,I)	- Sine (I = 1) and cosine (I = 2) of direction faced by IU-th unit.
ANYFOE	Called by FWDLIN whenever one of the sides (red or blue) has no units or op groups belonging to an engagement. Searches for a unit within scope of the engagement as defined by FINDFO (see Figure 5-95), in order to establish a FEBA.	Input:	DIREAX	- See ADJDIR.
			IFEBAB	- See ADJDIR.
			IFEBAR	- See ADJDIR.
			IFMUN(J)	- Forwardmost unit of operational grouping J.

			IOLDMVDT(I,IU)	- Stored movement data values for unit IU.
				I = 1 MVTCD(IU)
				I = 2 MVDT1(IU)
				I = 3 MVDT2(IU)
				I = 4 MVDT3(IU)

			MX	- NRU, if FWDLIN calculating blue FEBA, NUNIT, if red.
			MN	- 1, if FWDLIN calculating blue FEBA, NRUP1, if red.
			K	- Engagement being processed by FWDLIN.
		Output:	IXPHIB	- See ADJDIR. Set to -7999999 if no unit can be found in scope of engagement.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ANYFOE (Cont'd)			IXPHIR	- See ADJDIR. Also set to -7999999, if no unit can be found.
			LOCU	- Number of closest opposing unit within scope of engagement (see Figure 5-95). Set to zero if no unit within scope.
DEPLOC	Service routine called to give a movement code of 3 (deploying in an engagement) and proper movement data values to cause unit to deploy itself laterally relative to the forwardmost unit of the operational grouping, without moving closer to the FEBA. See Figure 5-96.	Input:	FECFAC(J)	- Frontage expansion-contraction factor for operational grouping J. Calculated by OGFRNT.
			IDUM	- Unit number for which calculations are to be performed.
			IFMUN	- See ANYFOE.

			IOGCDS(IU,I)	- Desired location of unit IU relative to the forwardmost unit (IFMUN) of its operational grouping.
				I = 1 distance behind
				I = 2 distance to side (plus is left)
				See Figure 5-96.

		Output:	MTCDOG	- See ADJDIR.
			MVTCD	- See ADJDIR.
			MVDT2(IU) MVDT3(IU)	- Movement data values 2 and 3 for unit IU. See Section 5.4 for a full explanation.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
DEPLOY	Examines all units in a given operational grouping to calculate a new engagement half-frontage. May also call OG2UNI to deploy each unit in op group.	Input:	JDUM	- Number of operational grouping for which calculations are desired.
			KDUM	- Number of engagement for which calculations are desired.
			KC	- Flag indicating whether op group is red (KC = 1) or blue (KC = 2).
		Output:	IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
DESTIN	Rotates and translates coordinates of a given point into a new coordinate system.	Input:	(IX1,IY1)	- (x,y) translations.
			(ST,CT)	- Sine and cosine of angle of rotation.
			(IX2,IY2)	- Coordinates of given point.
		Output:	(IX3,IY3)	- New coordinates of (IX2,IY2).
ENCTR	Determine if any red or blue units will come within engagement range of an enemy unit. If so, this information stored in movement code and movement data values. Old movement code, movement data saved for possible restoration.	Input:	ITRAV(IU)	- Travel code for unit IU. 0 = defunct air unit 1 = avoid engagement 2 = engage as necessary 3 = seek engagement 4 = active air unit
			MFIGHT(ITYPE,I)	- Range at which red (I = 1) and blue (I = 2) units of type ITYPE must engage enemy units. Input, in meters.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ENCTR (Cont'd)			MVTCD(IÜ)	- See subroutine ADJDIR.
			NGARNG(ITYPE,I)	- Same as MFIGHT, except range at which may engage. Input, in meters.
		Output:	IOLDMVDT	- See subroutine ANYFOE.
			MVTCD	- See subroutine ADJDIR.
			MVDT1	- See subroutine ADJDIR.
			MVDT2	- See subroutine ANYFOE.
			MVDT3	- See subroutine ANYFOE.
ENGDIR	Calculate direction given unit should face in given engagement.	Input:	DIREAX	- See ADJDIR.
			I	- Number of unit for which direction is desired.
			K	- Engagement in which unit is involved.
		Output:	(SOG,COG)	- Sine and cosine of direction to be faced. = (DIREAX(K,1), DIREAX(K,2)) for blue units. or = (-DIREAX(K,1),-DIREAX(K,2)) for red units.
FINDFO	Determines whether a given unit falls within a given range of a given engagement. (See Figure 5-95.)	Input:	II	- Number of unit for which calculations are desired.
			K	- Number of engagement being processed.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
FINDFO (Cont'd)			KRNG	- Range to use in calculations. Set by subroutine ANYFOE to NGARNG(IT,I) where IT = ITYPEU(II) I = 1 if II is red 2 if II is blue
			IFEBAB	- See subroutine ADJDIR.
			IFEBAR	- See subroutine ADJDIR.
			IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
			IXPHIB	- See subroutine ADJDIR.
			IXPHIR	- See subroutine ADJDIR.
		Output:	KK	- 1 if unit II not within scope of engagement K. 2 if unit II is within scope of engagement K. See Figure 5-95.
FRNTGS	Called once per time-step to calculate engagement half-frontages for each active engagement.	Input:	DIREAX	- See subroutine ADJDIR.
			IFEBAB	- See subroutine ADJDIR.
			IFEBAR	- See subroutine ADJDIR.
			MVDT1	- See subroutine ADJDIR.
			MVTCD	- See subroutine ADJDIR.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
FRNTGS (Cont'd)		Output:	IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
FWDLIN	Called once per time-step to establish new FEBA's, controlling operational groups, and unit distances from the FEBA for every active engagement. This data is used to determine whether units should become disengaged; if so, the old movement code and movement data values are restored. Any op group which no longer contains any active units is disengaged.	Input:	DIREAX	- See subroutine ADJDIR.
			IDWENG(J)	- Depth of operational grouping J when engaged (meters). Input.
			IOGSTAT(J)	- Status of operational group J. 0 means normal -1 means defunct
			NGARNG	- See subroutine ANYFOE.
		Output:	IFEBAB	- See subroutine ADJDIR.
			IFEBAR	- See subroutine ADJDIR
			IRFEBA(IU)	- Perpendicular distance from unit IU to closest enemy FEBA, in meters. Negative value indicates unit IU is behind enemy FEBA.
			ISTATU(IU)	- Status code for unit IU. -1 means air unit 0 normal 1 engaged, eligible to fire direct fire against enemy units in engagement.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
FWDLIN (Cont'd)			IXPHIB	- See subroutine ANYFOE.
			IXPHIR	- See subroutine ANYFOE.
			NCBOG(K)	- Number of blue operational grouping which is the "controlling" blue op group for engagement K. Set to zero if none.
			NCROG	- Same as NCBOG, except for red forces.
LATDST	Calculates perpendicular distance from a given unit to a given engagement axis.	Input:	IDUM	- Unit for which distance is to be calculated.
			KDUM	- Engagement for which distance is to be calculated.
			DIREAX	- See subroutine ADJDIR.
			IFEBAB	- See subroutine ADJDIR.
			IFEBAR	- See subroutine ADJDIR.
		Output:	IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
NEWENG	Called by subroutine ARRIVE to establish new engagements. All necessary engagement parameters are calculated.	Input:	IRB	- Flag indicating whether unit initiating engagement (unit which ARRIVE is processing) is red (IRB = 1) or blue (IRB = 2).
			IXYOG(J,L)	- X coordinate (L = 1) and Y coordinate (L = 2) of operational grouping J.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
NEWENG (Cont'd)			JB	- Flag indicating whether a blue unit (JB = 0) or a blue op group (JB = 1) is involved in the engagement.
			JR	- Same as JB, except for red forces.
			NOB	- Number of blue unit or op group (depending on value of JB) involved in engagement.
			NOR	- Same as NOB, except for red forces.
		Output:	IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
			IFEBAB	- See subroutine ADJDIR.
			IFEBAR	- See subroutine ADJDIR.
			IXPHIB	- See subroutine ADJDIR.
			IXPHIR	- See subroutine ADJDIR.
			MTCDOG	- See subroutine ADJDIR.
			MVTCD	- See subroutine ADJDIR.
			IOLDMVDT	- See subroutine ANYFOE.
NEWLOC	Calculate new coordinates reached by traveling a given distance in a given direction from a given point.	Input:	(IX1,IY1)	- (x,y) coordinates of initial point.
			(ST,CT)	- (sine,cosine) of direction of travel.
			IDIST	- Distance travelled.

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

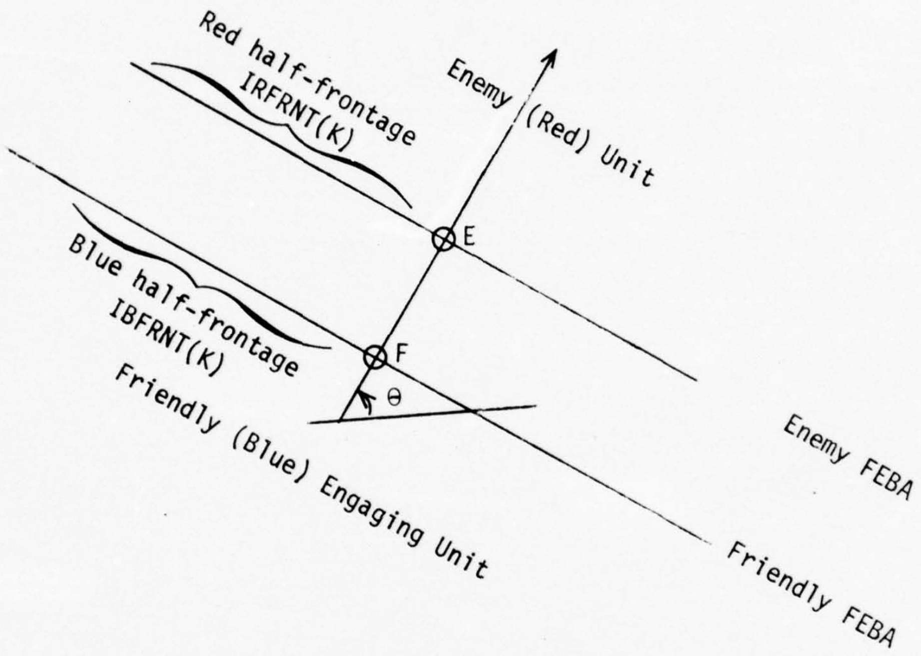
Subroutine	Description	Principal Inputs/Outputs		
NEWLOC (Cont'd)		Output:	(IX2,IY2)	- New coordinates reached.
NGARGN	Called to determine whether a unit (or its operational group) arriving at a destination relative to an engaged unit or operational grouping lies within the scope of that engagement, as defined by Figure 5-97.	Input:	IBFRNT	- See subroutine ADJDIR.
			IRFRNT	- See subroutine ADJDIR.
			DIREAX	- See subroutine ADJDIR.
			IFEBAB	- See subroutine ADJDIR.
			IXPHIB	- See subroutine ADJDIR.
			IXPHIR	- See subroutine ADJDIR.
			NGARNG	- See subroutine ENCTR.
			IDUM	- Unit for which calculations are to be made.
			K	- Engagement number for which calculations are to be made.
			ITYPDW(IT,I)	- Maximum distance in meters beyond end of enemy FEBA that an unengaged red (I = 1) or blue (I = 2) unit or operational grouping of type IT may be and still join this engagement rather than forming a new one. See Figure 5-97.
		Output:	KK	- Flag indicating whether unit IDUM may join engagement. =1 may not join =2 may join

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
OGFRNT	Whenever an operational grouping becomes engaged, OGFRNT is called to calculate the frontage expansion-contraction factor.	Input: HLFRN(J) - Normal half-frontage of operational grouping J, in meters. IEXTBL(L,M) - Expansion-contraction table containing a set of code words for modifying the frontages of operational groups entering an engagement. Table is packed, as follows: I-th table entry = IEXTBL(I,1) = UUVV IEXTBL(I,2) = WXXYYZ where $1 \le I \le 15$ UU = operational group number VV = unit type W = red (1) or blue (2) XX = unit type of controlling enemy op group in engagement. YY = unit type of controlling friendly operational group in engagement Z = expansion-contraction code 1 = adjust to fit enemy frontage

Table 5-45. Engagements Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
OGFRNT (Cont'd)		2 = adjust to fit friendly frontage The table is searched until a match is found between UU, VV, W, XX, YY and the existing situation. Z then gives the result. Note that zero in the table for UU, VV W, XX, or YY will match any corresponding entry in the existing situation. IDUM - Number of operational grouping for which calculations are to be performed. KDUM - Number of engagement for which calculations are to be performed. NCBOG - See subroutine ANYFOE. NCROG - See subroutine ANYFOE. Output: FECFAC - See subroutine DEPLOC.
WTHDRW	Initiates withdrawal of a unit which is engaged but a member of an unengaged operational grouping. Accomplished by setting direction of movement to reverse of engagement axis direction for unit.	Input: IDUM - Number of unit to be withdrawn. Output: MVTCD - See subroutine ADJDIR. PDIR - See subroutine ADJDIR.



F = location (midpoint) of friendly FEBA = (IFEBAB(K,1), IFEBAB(K,2))

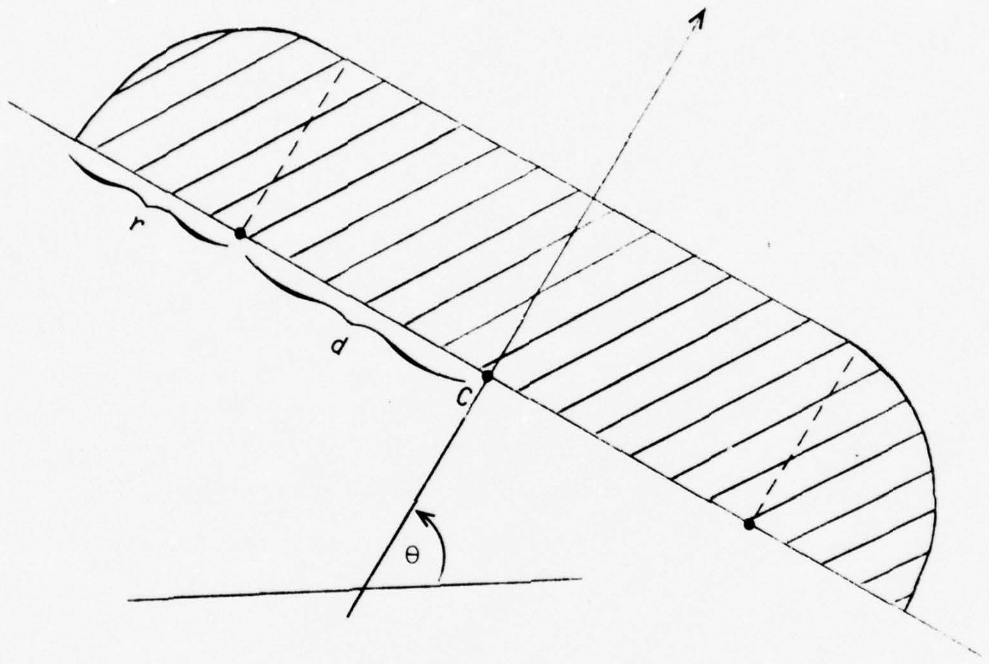
E = location (midpoint) of enemy FEBA = (IFEBAR(K,1), IFEBAR(K,2))

$\uparrow$ = engagement axis

$\begin{Bmatrix} \sin\theta \\ \cos\theta \end{Bmatrix}$ = direction of engagement axis from blue FEBA midpoint to red FEBA midpoint = (DIREAX(K,1), DIREAX(K,2))

K = engagement number

Figure 5-94. FEBA Definition



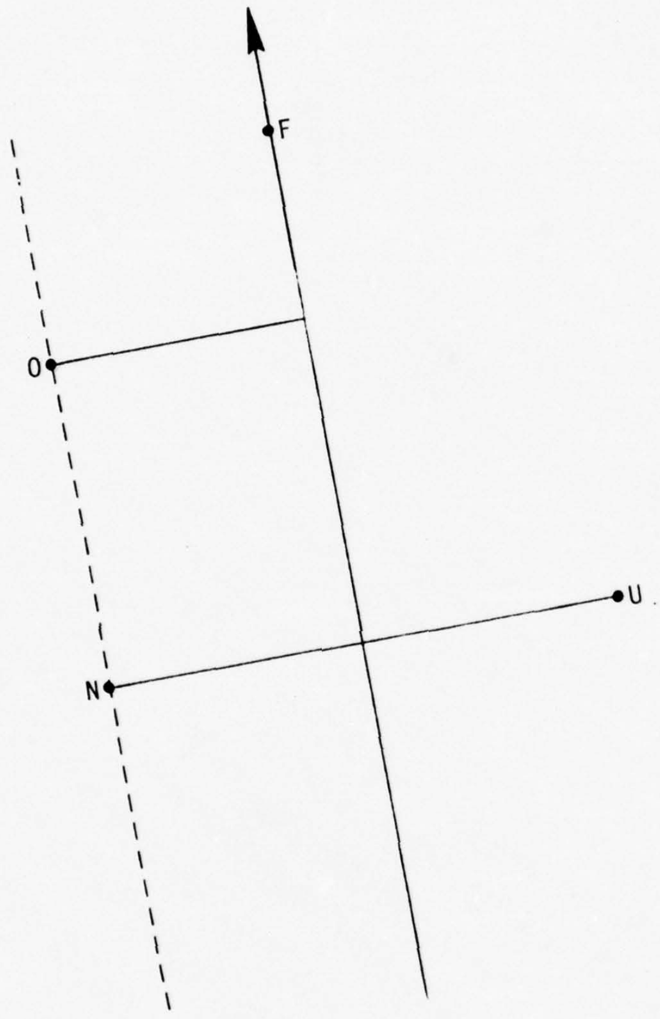
c = center location of friendly FEBA (IFEBAB)

d = half-frontage of friendly force (IBFRNT)

r = range from endpoint; also depth of scope area (NGARNG)

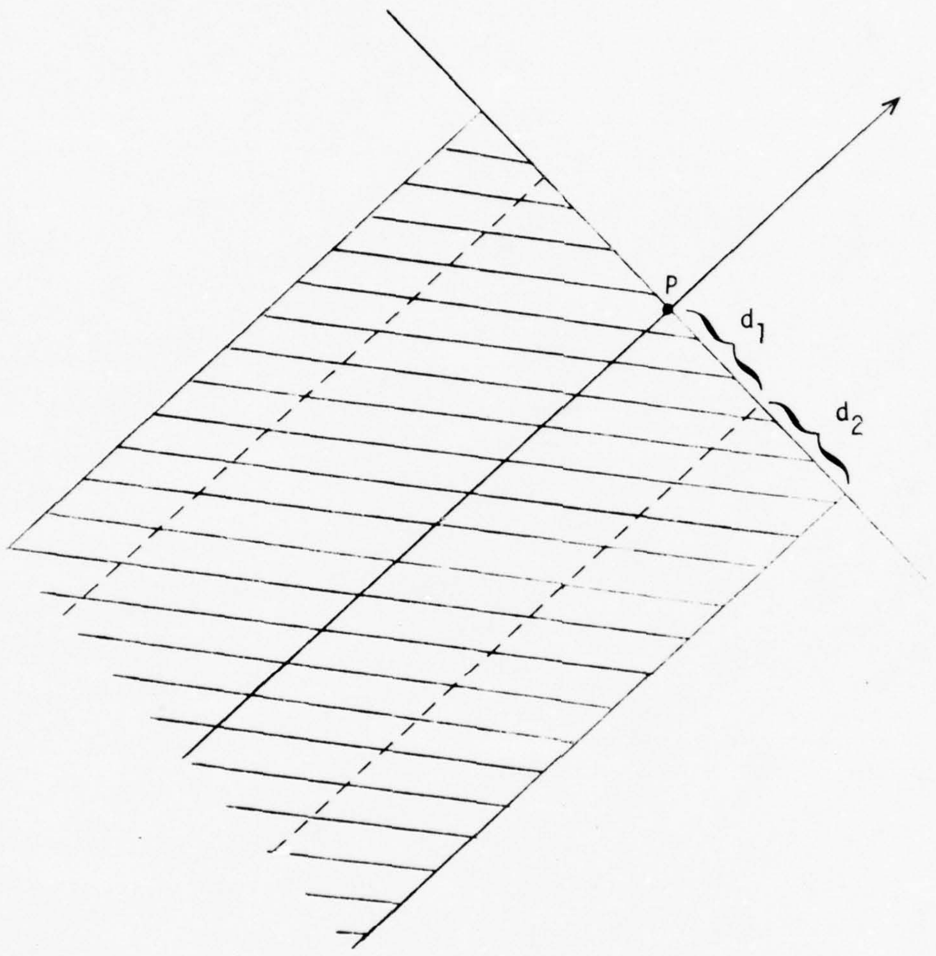
↑ = engagement axis defined by (sin θ, cos θ) (DIREAX)

Figure 5-95. Scope of the Engagement



U = present unit location of unit I
F = location of forwardmost unit in op group J
O = old deployment point of unit I
N = new deployment point of unit I
↑ = direction faced by op group J

Figure 5-96. DEPLOC Calculates a New Deployment Point for a Unit



P = center of opposing FEBA for engagement K

= (IFEBAB(K,1), IFEBAB(K,2)), if unit I is red
= (IFEBAR(K,1), IFEBAR(K,2)), if unit is blue

d_1 = opposing half-frontage of engagement K

= IBFRNT(K), if unit I is red
= IRFRNT(K), if unit I is blue

d_2 = maximum distance beyond end of opposing FEBA that a unit of the same type and side as unit I may join the existing engagement. If I is in an op group, then the type of its op group is used.

= ITYPDW(KT,1), if unit I is red
= ITYPDW(KT,2), if unit I is blue

Where KT = ITYPEU(I) if I does not belong to an op group, and KT = IOGTYP(INOG(I)) if it does.

Figure 5-97. NGARGN Calculates Whether a Unit Lies Within the Scope of an Engagement

directly toward the nearest enemy unit. Upon arrival at the enemy's location, the unit is instructed to form a new engagement if the (closest) enemy unit is not already engaged, or to join the engagement if the enemy unit is involved in a confrontation. In either case, the parameters governing unit movement and operational status are changed so that engagement maneuvers can be performed. Note that engagement initiation is triggered by the arrival of two opposing units; this is handled by the Ground Movement Module.

The creation of an engagement entails the definition of several attributes. These attributes dictate the formation and the relative positions taken by each unit involved in the engagement. In all, an engagement is comprised of six attributes. These are:

1) The direction of the engagement axis, DIREAX, calculated by ADJDIR with a call to ENGDIR.

2) The half-frontage of the blue and red forces, IBFRNT and IRFRNT, calculated by FRNTGS.

3) The X,Y coordinates of the center point of the blue and red forward edges of the battle area (FEBAs), IFEBAB and IFEBAR, calculated by FWDLIN.

4) The positions of the forwardmost operational grouping or unit for each side is expressed in a rotated coordinate system in which the positive X-axis is in the direction of the engagement axis; the resulting positions determine the center points of each FEBA. Calculated by FWDLIN, and stored in IXPHIB and IXPHIR.

5) The controlling operational groupings for each side, if they exist, NCBOG and NCROG, calculated by FWDLIN.

6) The area of close support for both blue and red forces, to be used to direct opposing artillery fire. Calculated by ADJDIR using input IENDP1 and IENDP2.

An engagement is uniquely defined by the numerical values assigned to each attribute. Engagement attributes (i.e., the numerical values assigned to them) change during the simulation according to battle conditions; this necessitates that the attributes be updated every time-step during the simulation.

5.6.1.1 Establishing the Forward Edge of the Battle Area (FEBA)

The forward edge of the battle area (FEBA) is the single most important concept used by the CATTS math model to simulate an engagement. When two opposing forces are in confrontation, two parallel reference lines, called FEBAs, are used by the movement and firing logic to perform engagement decisions and operations which simulate the battle. Specifically, units are deployed, or moved to certain deployment positions where they can utilize direct-fire weapons against one another.

Each FEBA has a definite location, orientation, and length associated with it. Figure 5-94 depicts the FEBAs for two opposing units. A FEBA is established by a point called the center location, and a frontage length. The center location is determined by the location of the controlling (i.e., forwardmost) operational grouping, if it exists. If there is no controlling operational grouping, then the unit belonging to the engagement which is nearest to the enemy force has its location taken as the center location of the FEBA. The word "forwardmost" and the phrase "nearest to the enemy force" are synonymous. This determination is made by examining unit locations in a rotated coordinate system having the positive X-axis coincide with the direction of the engagement axis. When the center location for each side has been established, the vector drawn from the blue center to the red center redefines the engagement axis. Note that by convention, the orientation of the engagement axis is always taken to be a directed segment originating at the blue center and terminating at the red center. Thus, the direction of the engagement axis is uniquely specified. Once the axis vector has been constructed, a pair of parallel line segments, one passing through each center location, perpendicular to the engagement axis, form the frontages for the respective sides participating in the engagement. The length of the frontage is equal to twice the half-frontage for the respective forces. The half-frontage is either half the unit width if only a single unit is involved, or in the case where an operational grouping is involved, the lateral distance determined by the unit of the group having the largest perpendicular distance from the engagement axis plus half of that unit's width. Five of the first six attributes listed above are used to determine ιne FEBA. This gives an indication of the relative importance of the FEBA concept in the simulation of engagements.

5.6.1.2 Ground Fire During Engagements

Engagements affect the use of support and direct fire armaments. For automatic support-fire allocation, an engagement has close support and interdictory regions defined. These regions are rectangular areas whose sides are determined by 1) the larger of the two engagement frontages, and 2) a given depth measured either from the friendly FEBA or enemy FEBA, depending upon the value of the fire support weapon code. Any opposing unit within the close support region is fired at from mode 3 or 4, the close support modes. Any opposing unit within the interdictory region is fired at from mode 5. A more detailed explanation of support-fire weapons is given in Section 5.4.

Engagements allow units to utilize their direct-fire armaments. To be eligible to do such, the engaged unit must satisfy certain criteria. The main requirement involves the distance between the unit and the enemy FEBA (see equation 8). This distance must not exceed the unit's pre-determined engagement range. Other unit considerations, such as maneuver control status, location relative to the controlling operational grouping if one exists, etc., are examined before eligibility to discharge direct-fire weapons can be established. These details are discussed in subsequent sections. In general, an engagement will involve opposing forces focusing their attentions primarily toward one another, especially to utilize their direct-fire weapons against one another. Two opposing units, however, are not prohibited from firing at each other when either or both are unengaged. Likewise, two opposing units in different engagements are not prohibited from firing at each other.

5.6.2 Operation

The determination as to when an engagement is to be started is made by the Ground Movement Module. Recall that during each time-step, a new location is scheduled for a unit based upon the unit's current location, direction of movement, and speed. However, before the move is conducted, a check is made to ensure that the unit will neither encounter an obstacle obstruction nor confront an enemy unit. Both occurrences will prevent the unit from achieving its scheduled move. In the latter instance, several criteria are evaluated to determine whether a new engagement can be initiated. The most important criterion involves the distance to the nearest enemy unit. If the several criteria are satisfied, the unit is instructed to move toward the enemy unit (movement code 14). A new destination (i.e., the location of the enemy unit) is specified, thus modifying the direction of movement for the unit. This, in turn, causes a revised location to be scheduled for the unit. But before the revision takes place, the unit's previous movement code and movement data are saved. This allows the unit's previous movement status to be reinstated in the event that an engagement is not formed, or upon eventual termination of the engagement. The changing of unit movement code to 14 in order to pursue an enemy unit as a result satisfying certain criteria signifies the first step of engagement initiation.

All unengaged moving units are inspected, first red, then blue to determine if the moves scheduled during the current time-step will satisfy certain criteria necessary to form engagements with enemy units. Units of the red army, designated first as the weapon units, are moved against target units of the blue army; then their respective roles are interchanged. A given weapon unit is evaluated along with each and every target unit. The target unit must be a nondefunct ground unit. The weapon unit is also required to be a nondefunct ground unit. It must be moving in an unengaged condition (i.e., movement codes 5, 6, 8, 9, 10, 11, 12, 13, 14, or 15). With respect to obstacles, it must not be in the process of traversing through an obstacle. The first duty of a unit caught within an obstacle is to clear the obstacle; then it is permitted to pursue an enemy unit.

5.6.2.1 Engagement Criteria Between Weapon and Target Units

The most important criteria governing the formation of a new engagement are: 1) visual detection of the target unit by the weapon unit, and 2) the distance separating the two units. Before an engagement can be initiated, visual contact by the weapon unit must have been established with the target unit. If no such detection is made, the weapon unit will immediately ignore the target unit and proceed to consider the next target unit. Engagement initiation cannot occur with unseen target units.

The detection decision is based upon the values in table IFIREFA, as stored by subroutine DETECT (see Section 5.3.1).

However, once detection is made, the distance between the opposing units is determined. The weapon unit has a characteristic data input engagement range NGARNG, determined by unit type and side, that specifies the distance at which it may initiate an engagement with an enemy unit. It also has another characteristic range, also pre-defined by data input MFIGHT, according to unit type, which specifies the range at which it must initiate an engagement with an enemy unit. This more restrictive range is called the must-fight range; any target located within the weapon unit's must-fight range will be automatically considered for engagement. This consideration is made regardless of the intents and maneuver control statuses of the units involved. In other words, the target unit is so near the weapon unit (perhaps 50 meters), that a new engagement must be started. This will happen unless the weapon unit encounters yet another target unit that is even closer during subsequent processing.

Three cases exist with respect to a given weapon unit's engagement and must-fight ranges. They are as follows:

1) Target units may be located beyond the weapon unit's engagement range.

2) Target units may be located within the weapon unit's engagement range but beyond its must-fight range.

3) Target units may be located within the weapon unit's must-fight range.

The first case is handled easily. Target units located beyond the weapon unit's engagement range are ignored. No action is initiated; the weapon unit immediately proceeds to consider the next target unit.

The second case requires a detailed examination of several unit attributes. Target units which are within the weapon unit's engagement range, but beyond the must-fight range may or may not be eligible to originate an engagement. The verdict depends upon certain attributes:

1) Whether the unit is a member of an operational grouping

2) Intent of the unit (1 = avoid engagement, 2= engage as

necessary, 3 = seek engagement)

3) Whether the unit is engaged or unengaged

An engagement may be started if the weapon and target units have attributes which coincide with one of the sets of attributes described in Table 5-46. The logic involved in determining eligibility to initiate engagement is illustrated in Figure 5-98. The numeral atop each decision box traces out the logic path corresponding to the set of attributes detailed in Table 5-46.

Among all eligible target units, the one that is nearest the weapon unit is chosen to participate in forming the new engagement. The weapon unit will have its movement code changed to 14 (moving to a point relative to an enemy unit) so that its objective is now to pursue the nearest target unit. Note that before the change takes place, the weapon unit's old movement code and movement data are saved. This precaution is taken in the event that engagement initiation is aborted.

The weapon unit will travel under movement code 14 until it arrives at a point where it will confront the enemy unit. At this point, both weapon and target units are released from maneuver control if either were under such control. This will allow the built-in movement logic to set up the values to be assigned for each engagement attribute. If the enemy is already involved in an engagement, a new engagement is not started. Instead, the weapon unit will join the existing engagement. If the weapon unit is a member of an operational grouping, the other members of the grouping are made to deploy into pre-determined positions relative to the weapon unit. They too will join the engagement.

If the enemy unit is unengaged, a new engagement is started. The following actions are taken to assign values to attributes which uniquely define the engagement:

1) Determine red and blue FEBAs (IFEBAB, IFEBAR, IXPHIB, IXPHIR).

2) Compute engagement axis vector from blue FEBA center to red FEBA center (DIREAX).

3) Deploy units relative to operational grouping in groupings which are moving toward an engagement intending to deploy on arrival.

Table 5-46. Weapon-Target Attributes Required For Engagement Initiation

ATTRIBUTES / SET NO.	WEAPON UNIT IN OP.GROUP.	TARGET UNIT IN OP.GROUP.	INTENT OF WEAPON UNIT	INTENT OF TARGET UNIT	WEAPON UNIT ENGAGED	TARGET UNIT ENGAGED
1	NO	YES	3	1,2,OR 3	NO	YES OR NO
2	NO	NO	2	2 OR 3	NO	NO
3	NO	NO	3	1,2,OR 3	NO	YES OR NO
4	NO	NO	1	3	NO	NO
5	YES	YES	2	2 OR 3	NO	NO
6	YES	YES	3	1,2,OR 3	NO	YES OR NO

<u>Intent Code</u>:

1 = avoid engagement

2 = engage as necessary

3 = seek engagement

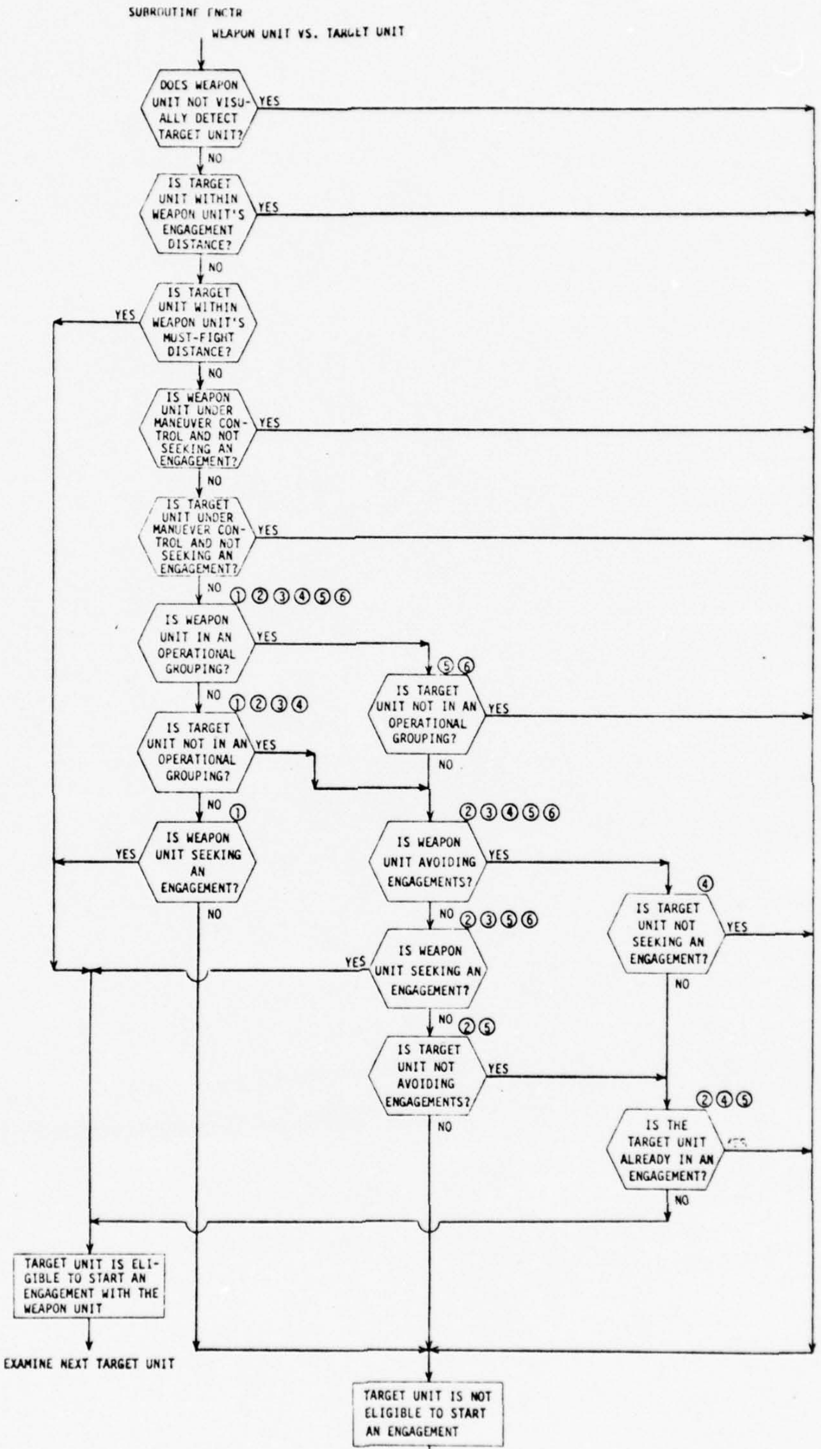

SUBROUTINE ENCTR
WEAPON UNIT VS. TARGET UNIT
DOES WEAPON UNIT NOT VISUALLY DETECT TARGET UNIT?
IS TARGET UNIT WITHIN WEAPON UNIT'S ENGAGEMENT DISTANCE?
IS TARGET UNIT WITHIN WEAPON UNIT'S MUST-FIGHT DISTANCE?
IS WEAPON UNIT UNDER MANEUVER CONTROL AND NOT SEEKING AN ENGAGEMENT?
IS TARGET UNIT UNDER MANEUVER CONTROL AND NOT SEEKING AN ENGAGEMENT?
IS WEAPON UNIT IN AN OPERATIONAL GROUPING?
IS TARGET UNIT NOT IN AN OPERATIONAL GROUPING?
IS WEAPON UNIT SEEKING AN ENGAGEMENT?
IS WEAPON UNIT AVOIDING ENGAGEMENTS?
IS TARGET UNIT NOT SEEKING AN ENGAGEMENT?
IS TARGET UNIT NOT AVOIDING ENGAGEMENTS?
IS THE TARGET UNIT ALREADY IN AN ENGAGEMENT?
YES
NO
①②③④⑤⑥
TARGET UNIT IS ELIGIBLE TO START AN ENGAGEMENT WITH THE WEAPON UNIT
EXAMINE NEXT TARGET UNIT
TARGET UNIT IS NOT ELIGIBLE TO START AN ENGAGEMENT
EXAMINE NEXT TARGET UNIT

Figure 5-98. Logic Determining Eligibility of Engagement Initiation

4) Determine which side controls the engagement (NCBØG, NCRØG).

5) Calculate engagement frontage IBFRNT, IRFRNT of controlling side from either operational grouping deployment or unit width.

6) Calculate engagement frontage of opposing side by expanding or contracting the opposing side's normal frontage, if applicable, or simply calculate normal frontages from operational grouping deployment or unit width.

7) Deploy units in controlling operational groupings on both sides of the engagement.

5.6.2.2 Updating and Maintaining Engagements

All values describing the entities which comprise an engagement are updated once per time-step. The updates are accomplished in four distinct steps. The first step deals with one-sided engagements. The axes of single-sided engagements are re-defined each time-step; units and operational groupings involved in such engagements are re-deployed to correspond to the new axes. The second step recomputes the location of each nondefunct operational grouping and determines the unit within each operational grouping which is farthest forward in the direction of attack. The third step examines each engagement to update the following:

1) FEBA locations for the red and blue forces involved (IFEBAR, IFEBAB).

2) The new red and blue controlling operational groups if they exist for the engagement (NCROG, NCBOG).

3) The distance of each unit involved in the engagement from its respective enemy FEBA (IXPHIR, IXPHIB).

Finally, the fourth step recalculates the half-width of each engagement frontage (IBFRNT, IRFRNT).

5.6.2.2.1 Updating One-sided Engagements

The engagement axis of a one-sided engagement is updated each time-step. The prime objective is to keep the attacking force aimed toward the center of greatest immediate enemy threat. A radius of responsibility about the attacking force's FEBA center is established. This radius is determined by the distance between the centers of the red and blue FEBA's plus the close support region width for the attacking force. The center of greatest immediate threat is taken to be the center of mass of all opposing units having the following properties:

1) The opposing unit is not part of this engagement, or is unengaged.

2) The distance from the unit to the attacking force's FEBA center is less than the radius of responsibility.

3) The opposing unit is in front of the attacking force's FEBA.

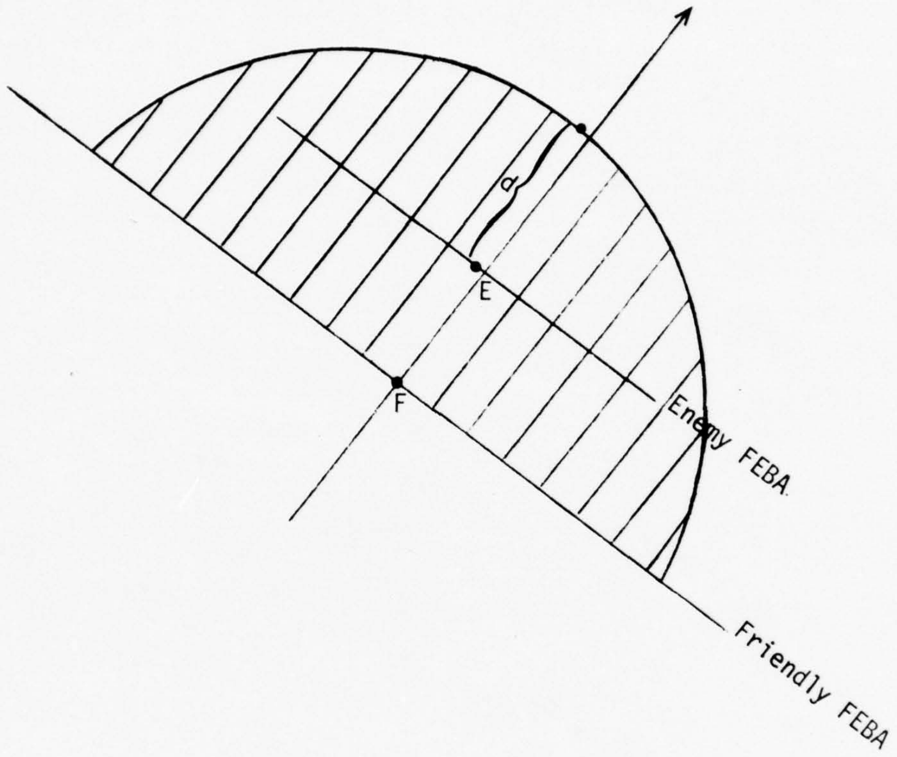
These properties will encompass possible enemy target units which are located within a semi-circular area about the attacking force's FEBA center. This is illustrated by Figure 5-99. The direction of the engagement axis is established by drawing the vector between the center of mass and the attacking force's FEBA center (the orientation of the vector, of course, must be from blue to red, regardless of the color of the respective forces).

Once the direction of the engagement axis has been updated, all member units of the attacking force have their directions of movement made to point at the enemy's center of mass. If the attacking units make up operational groupings, then these operational groupings have their directions of movement similarly modified. Furthermore, if any of those operational groupings are deployed, each unit in the operational groupings is redeployed according to the new engagement axis direction. The direction of a one-sided engagement axis is recomputed each time-step to guide the attacking force toward the area of greatest enemy threat.

5.6.2.2.2 Updating Locations of Operational Groupings

The locations of all nondefunct operational groupings are recomputed each time-step. This must be done because during engagements, units within operational groupings are moving toward their deployment positions. In addition, the forwardmost unit (i.e., closest to the enemy) in an operational grouping may change during the simulation according to battle conditions.

The update procedure begins by determining the location of the most advanced unit in the operational grouping. To achieve this, the coordinates of each member unit in the operational grouping are transformed to a



F = center of friendly FEBA (IFEBAB)

E = center of enemy FEBA (IFEBAR)

d = depth of close support region (IENDP1 or IENDP2)

$\uparrow$ = engagement axis (DIREAX)

Figure 5-99. Area of Responsibility to Determine Greatest Immediate Threat in One-sided Engagements

rotated coordinate system whose positive x-direction coincides with the direction of movement of the operational grouping. This transformation is represented by the matrix in equation 3. The forwardmost unit is determined by examining the transformed x-coordinate of each unit in the operational grouping. The unit having the largest x-coordinate value is the unit which is most advanced. Identifying the forwardmost unit is important because the positioning, movement, and conduct of the non-lead units of the group are affected by the actions of the lead unit. For instance, deployment positions are made with respect to the forwardmost unit. Another example involves movement speed; members of an operational grouping cannot be moving faster than the forwardmost unit.

The location of an operational grouping is determined by one of two methods, depending upon its movement code. First of all, operational groupings having movement codes 4 or 16 need not have their locations updated since their member units are all deployed and waiting for other forces to deploy. These operational groupings retain their previous location. Operational groupings having movement codes 5 through 14 have their locations determined by the locations of their forwardmost units. Finally, operational groupings which are engaged and moving (movement codes 1, 2 or 3) or not engaged but in the process of deploying (movement code 15), have their locations computed in the rotated system. Specifically, the transformed x-coordinate of the forwardmost unit, and the mean transformed y-coordinate of the units in the grouping determine the new location (in the rotated system) of the operational grouping. The coordinates of the new location are transformed back into the original Cartesian system (see equation 9).

5.6.2.2.3 Updating FEBA Centers

The center location of each side's FEBA is recomputed every time-step. Also the controlling operational grouping, if they exist, for each side is determined. Furthermore, the distance from the enemy FEBA to each unit participating in the engagement is recalculated (see equation 8). This distance is used to determine whether the unit is eligible to discharge any direct-fire weapons; it is also used to determine whether it should become disengaged. A unit which becomes disengaged has its old movement status (i.e., pre-engaged movement code and data) restored. An operational grouping which no longer contains any engaged units is disengaged.

The controlling operational grouping for a given side, if it has one, is the one which belongs to the engagement and contains a unit farthest forward in the direction in which it is engaged. This determination is made by examining the transformed x-coordinates of all units which are members of the operational grouping. The transformed x-coordinate is obtained by expressing the unit's position coordinates in a rotated coordinate system whereby the positive x-direction is in the direction of movement of the operational grouping. The unit possessing the greatest transformed x value determines two important engagement attributes:

1) The operational grouping of which it is a member becomes the controlling operational grouping.

2) The X,Y location of this unit (upon transformation back to the standard Cartesian coordinate system), becomes the updated center point of the FEBA.

Thus, the location of the controlling operational grouping for a given side determines that side's engagement FEBA.

As the discussion above indicates, the location of the center point of a FEBA is very much dependent upon the location of the controlling operational grouping. However, an engagement may involve opposing forces, in which one or both sides will lack a controlling operational group. Thus, when updating FEBA locations, three cases may arise:

1) Both sides have controlling operational groupings.

2) One side in the engagement lacks a controlling operational grouping.

3) Both sides lack controlling operational groupings.

The first case is handled easily. As indicated by the previous paragraph, the controlling operational grouping for each side is determined. The X,Y locations of these groupings become the updated center locations for the respective FEBAs.

In the second case, the side lacking a controlling operational grouping is searched for an engaged unit lying nearest the enemy FEBA.

Note that since the opposing side has a controlling operational grouping, its FEBA location is uniquely determined by the controlling operational grouping's location. If the search yields an engaged unit which is:

1) Participating in the engagement

2) Farthest advanced toward the enemy FEBA

then a new FEBA location can be established for its side. The updated FEBA location is given by the point obtained from the perpendicular projection of the unit's location onto the engagement axis. This is illustrated by equation 5. If there is no such unit participating in the engagement, then the search is extended to find the closest engaged unit not in this engagement but within the scope of the engagement. The scope of the engagement is a band-like region in front of the established FEBA as illustrated by Figure 5-95. The nearest engaged unit (that is not part of this engagement) relative to the enemy's FEBA, lying within the shaded region called the scope of the engagement, determines the new FEBA location. As before, the point generated by the perpendicular projection of this unit's location onto the engagement axis, becomes the updated FEBA center. If the extended search fails to find an engaged unit, a new FEBA center cannot be determined. This will cause the engagement to terminate; all units participating in the engagement, but not under maneuver control, will have their movement codes and movement data restored to pre-engagement values.

In the third case, when an engagement lacks controlling operational groupings on both sides, the respective FEBA centers are determined by finding for each side, the unit belonging to the engagement and farthest forward in its direction of movement toward the enemy. If such a unit exists for a given side, that side's new FEBA center is determined by that unit's location. If neither side has such a unit, the engagement is vacuous and is terminated immediately. On the other hand, if only one side has a unit belonging to the engagement, the opposing side is searched for an engaged unit which is not part of the engagement but lying within the scope of the engagement. Again, the scope of an engagement is the band-like area in front of the established FEBA (see Figure 5-95). The nearest engaged unit lying within the scope of the engagement is chosen to determine the new FEBA location. Specifically, the point obtained by the perpendicular projection of the chosen unit's location onto the engagement axis becomes the updated FEBA center.

5.6.2.2.4 Updating Engagement Frontages

The frontages of each active engagement is updated every time-step. It is convenient to talk about half-frontages since the model calculates the half-widths of each frontage per engagement. The half-frontage for a given side is computed by considering all its non-defunct engaged units which are eligible to fire direct-fire weapons. For each such unit, the perpendicular distance from its location to the engagement axis is determined. If the unit is deploying in the engagement (movement code 3), the perpendicular distance is taken from the unit's deployment point to the engagement axis. In either case, the resulting perpendicular distance is added to one-half the width of the unit. This overall distance represents the half-width of the frontage produced by the unit. After all such distances have been computed for one side, the largest distance is chosen to be that side's new engagement half-frontage . The updated frontage is obtained by doubling the distance given by the new half-frontage.

Note that if one side of an engagement consists of a single unit, the half-frontage for this side is given by half the unit width. In other words, its engagement frontage will span the entire width of the lone unit. Generally an engagement consists of two frontages, one for the red forces and one for the blue forces. However, there are instances when an engagement will contain only one. If an already engaged operational grouping or unit becomes involved, as a non-initiator, in another engagement (i.e., it is being attacked by another enemy force thus forming a one-sided engagement), its engagement frontage is set to zero indicating a non-existent frontage.

5.6.2.3 Determining Direct-Fire Eligibilities

Each unit currently involved in an engagement is examined to determine whether it is eligible to fire its direct-fire weapons. A unit which is a member of a controlling operational grouping, or an engaged unit whose side is without a controlling operational grouping will be permitted to discharge its direct-fire weapons if at least one of the following conditions hold:

1) The distance between the unit and the enemy FEBA is less than the engagement range for that unit

2) The unit is moving under maneuver control

3) The unit is a member of an operational grouping

If a unit belongs to an operational grouping other than the controlling one, it must lie forward of the control grouping's rear boundary and satisfy at least one of the above three conditions in order to be eligible to use its direct-fire arms. Such a unit which does not lie forward of the control grouping's rear boundary is immediately eliminated from direct-fire eligibility.

5.6.2.4 Determining Disengagement

During each time-step, the range from an engaged unit to its enemy FEBA is computed. This range is used to determine the unit's eligibility to fire direct-fire arms as discussed above. The range is also used to determine whether an engaged unit will become disengaged. Specifically, an engaged unit which satisfies all of the following three criteria will become disengaged:

1) The distance between the unit and the enemy FEBA exceeds the engagement range for the unit

2) The unit is not moving under maneuver control

3) The unit is not a member of an operational grouping

Disengagement entails a restoration of pre-engagement movement code and movement data values.

Engaged operational groupings are examined each time-step to ascertain whether disengagement is appropriate. If an engaged operational grouping contains no units which are within engagement range of their enemy FEBAs, disengagement is in order. Specifically, the old pre-engagement movement code and movement data values of the grouping's forwardmost unit are used to update the grouping's movement status. When an operational grouping is made unengaged each and every member unit has its old pre-engagement movement code and movement data values restored. Thus, disengagement consists in changing the movement status of the operational grouping as well as the movement statuses of each unit contained in the grouping.

5.6.3 Assumptions and Data Sources

The assumptions used in constructing the Engagements Module are:

1) The most important considerations as far as whether a given unit will join an engagement or initiate a new engagement are: visual detection and distance. Other considerations include: maneuver control status, intent of unit, whether it is a member of an operational grouping, and unit movement code.

2) In modeling an engagement, the most important attributes are used to construct parallel reference lines called FEBAs which are used to govern ground movement and ground fire decisions.

3) Each side in an engagement may be made up of a:

   a) Single unit

   b) Group of individual units (not in an operational grouping)

   c) Single operational grouping

   d) Set of operational groupings

In cases (c) and (d), a controlling operational grouping exists and is determined. *Controlling operational groupings determine the characteristics* of that side's FEBA.

4) The unit which is forwardmost towards the enemy establishes the FEBA location for that unit's side. If the unit is a member of an operational grouping, then that operational grouping becomes the controlling operational grouping for that side.

5) Units change from an unengaged movement code to an engaged movement code only by automatic means (built-in software); i.e., they cannot be directed to make such a change interactively through maneuver control.

6) Units which are members of an operational grouping have predetermined (by user input) deployment locations relative to a point defined by one of the following:

   a) The operational grouping's location as determined by its forwardmost unit,

   b) The operational grouping's location as determined by the x-coordinate of the forwardmost unit and the mean y-coordinate of all units in the grouping.

The choice is dictated by whether the operational grouping is actually deployed (b) or is moving intending to deploy (a).

7) An operational grouping is not considered deployed and in an engagement until every member unit has reached its pre-determined deployment location. As each unit of the operational grouping reaches its deployed location, its movement code is automatically changed to 4 unless specifically directed to another unengaged value (i.e., by maneuver control). When all units of an operational grouping have deployed, the movement code of each is changed automatically to 1 unless the movement code of the operational grouping has been changed to some value other than 3 or 4 in the meantime.

8) If an operational grouping or unit is engaged when an opposing force has arrived (movement codes 10 or 14), the arriving force will usually join the engagement; it may, however, elect to start a new engagement depending on its position relative to the friendly FEBA and other enemy units. An arriving unit which confronts an unengaged opponent will start a new engagement.

9) Engaged units are eligible to discharge their direct-fire arms against the enemy if they are within certain range of the enemy FEBA. Specifically, the following holds:

   a) Units not part of an operational grouping must be within engagement range of the enemy FEBA.

   b) Units which are members of an operational grouping but not in the controlling operational grouping must lie forward of the rear boundary of the control group.

   c) Units which are members of the controlling operational grouping are automatically eligible to use their direct-fire arms.

10) It should be noted that a unit can be assigned to only one engagement at a time, but it could actually be firing simultaneously on units involved in other engagements. It can also fire upon unengaged enemy units.

11) A unit becomes disengaged when its range from the enemy FEBA is beyond the pre-defined engagement range for that type of unit. Also, an operational grouping becomes disengaged when none of its member units are within engagement range of the enemy FEBA.

Note: All assumptions and data sources for the Engagement Module are based on the following:

MAFIA User's Manual, United States Army Combat Developments Command, 1969.

Engineering judgement at TRW.

5.6.4 Equations

The important tactical and physical mathematical equations used by the Engagements Module are:

1) Determination of new X,Y location for the weapon unit.

Given the following quantities:

a) The X,Y location (IXY) of the weapon unit: (X_1,Y_1),

b) The direction of movement (PDIR) of the weapon unit: $(\sin\theta, \cos\theta)$,

c) The distance D to be traversed by the unit while moving at speed R (ROMU) during the current time-step: $D = RT$, where $T = 1$ minute,

The new X,Y location for the unit (X_2,Y_2) is given by the following equation:

$$X_2 = X_1 + D\cos\theta$$

$$Y_2 = Y_1 + D\sin\theta$$

2) Determination of the distance between the weapon unit and a target unit.

Given the following quantities:

a) The X,Y location (IXY) of the weapon unit: (X_W,Y_W),

b) The X,Y location (IXY) of the target unit: (X_T,Y_T),

the distance D between the weapon unit and the target unit is given by the following equation:

$$D = \sqrt{(X_W - X_T)^2 + (Y_W - Y_T)^2}$$

The distance D is compared with the weapon unit's engagement (NGARNG) and must-fight (MFIGHT) ranges to determine whether initiation of a new engagement is possible.

3) Determining the forwardmost unit (toward the enemy) in the direction of movement (which coincides with the direction of the engagement axis when the unit is engaged).

Given the following quantities:

a) The X,Y location (IXY) of the unit: (X,Y),

b) The direction of movement (PDIR) of the unit: $(\sin\theta, \cos\theta)$,

c) The matrix of transformation used to express the X,Y location of the unit in a rotated coordinate system having the positive X-axis coinciding with the direction of movement is:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

The unit location, expressed in terms of the rotated system is given by (X^1, Y^1) as follows:

$$(X,Y)\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = (X\cos\theta + Y\sin\theta\ ,\ -X\sin\theta + Y\cos\theta\ ) = (X^1, Y^1)$$

Specifically, $X^1 = X\cos\theta + Y\sin\theta$

$Y^1 = -X\sin\theta + Y\cos\theta$

Every unit per side involved in the engagement has its location expressed in the rotated system. The unit having the largest abscissa value (i.e., the largest X^1 value) is determined to be unit furthermost advanced towards the enemy.

4) Determining the direction of the engagement axis.
Given the following quantities:

a) The FEBA center (IFEBAB) of the blue force: (X_B, Y_B)

b) The FEBA center (IFEBAR) of the red force: (X_R, Y_R)

The direction (DIREAX) of the engagement axis (A_1, A_2) is given by:

$$A_1 = \frac{Y_R - Y_B}{D}$$

$$A_2 = \frac{X_R - X_B}{D}$$

where: $D = \sqrt{(Y_R - Y_B)^2 + (X_R - X_B)^2}$

Note that the direction of the engagement axis, by convention, is always taken from the blue FEBA center to the red FEBA center. This implies that with respect to the red force, the direction of movement in the engagement (i.e., from red FEBA center to blue FEBA center) is given by $(-A_1, -A_2)$.

5) Determine FEBA center location by perpendicular projection onto the engagement axis of the forwardmost unit not in the engagement, but located within an area called the scope of the engagement.

Given the following quantities:

a) The FEBA center location (IFEBAR) of the enemy force: (X_E, Y_E)

b) The location (IXY) of unit lying within the scope: (X,Y)

c) The direction (DIREAX) of the engagement axis: $(\sin\theta, \cos\theta)$

The FEBA center (IFEBAB) location (X_F, Y_F) for the friendly unit is given by:

$$X_F = \frac{B_2 - B_1}{A_2 - A_1}$$

$$Y_F = A_1 X_F + B_1$$

where:

$$A_1 = \frac{\sin\theta}{\cos\theta} \qquad A_2 = \frac{-\cos\theta}{\sin\theta}$$

$$B_1 = Y_E - A_1 X_E \qquad B_2 = Y - A_2 X$$

6) Determination of engagement half-frontage.

Given the following quantities:

a) The FEBA center (IFEBAB) location: (X_F, Y_F)

b) The direction (DIREAX) of the engagement axis: $(\sin\theta, \cos\theta)$

c) The location (IXY) of the unit involved in the engagement: (X,Y)

d) The width (IUWID) of the unit: W

The half-frontage (IBFRNT) D for a given unit is obtained from the following equation:

$$D = \frac{W}{2} + \sqrt{(X-X_o)^2 + (Y-Y_o)^2}$$

where:

$$X_o = \frac{B_2-B_1}{A_2-A_1}$$

$$Y_o = A_1X_o + B_1$$

$$A_1 = \frac{\sin\theta}{\cos\theta} \qquad A_2 = \frac{-\cos\theta}{\sin\theta}$$

$$B_1 = Y_F - A_1X_E \qquad B_2 = Y-A_2X$$

Each and every unit on one side of the engagement has a half-frontage distance D associated with it. This distance is the perpendicular from the unit location to the axis of engagement plus half the unit width. The unit producing the largest such distance D determines that side's engagement half-frontage.

7) Determining close-support fire region during the engagement.

Given the following quantities:

a) The half-frontage (IRFRNT, IBFRNT) of the red and blue forces participating in the engagement: H_R, H_B,

b) The depth (IENDP1, IENDP2) of the region in an engagement, in which close-support targets may be assigned: D,

The close-support fire region A is given by the rectangular area described by the following equation:

$A = (2 \times H) \times B$

where: $H = \min \left\{ H_R, H_B \right\}$

8) Computation of range (IXPHIR, IXPHIB) to enemy FEBA.

Given the following quantities:

a) The unit location: (X,Y)

b) The abscissa value of the center location of the enemy FEBA is a rotated coordinate system whereby its positive x-axis coincides with the direction of the engagement axis:

ϕ_R (red FEBA), ϕ_B (blue FEBA)

c) The direction of the engagement axis: $(\sin\theta, \cos\theta)$

The range R from a given unit to the enemy FEBA is given by the following equations:

$$R = \begin{cases} -\phi_R - (X\cos\theta + Y\sin\theta), & \text{if enemy FEBA is red} \\ -\phi_B + (X\cos\theta + Y\sin\theta), & \text{if enemy FEBA is blue} \end{cases}$$

9) Determining the location of an operational grouping.

Given the following quantities:

a) The X,Y coordinate values (IXY) of each member unit in the Jth operational grouping: (X_{J_i}, Y_{J_i})

b) The total number of units comprising the Jth operational grouping: N (i.e., $1 \le i \le N$)

c) The location of the forwardmost unit (IFMUN(J)) for the Jth operational grouping: (X_{J_f}, Y_{J_f}) $(1 \le f \le N)$,

d) The direction of movement of the Jth operational grouping: $(\sin\theta, \cos\theta)$

The location (IXYOG) of the Jth operational grouping (X_J, Y_J) is determined in one of two ways:

(1) If the Jth operational grouping is not engaged and not actually deployed, its location is given by its forwardmost unit:

$$X_J = X_{J_f}$$

$$Y_J = Y_{J_f}$$

(2) If the Jth operational grouping is engaged, or deployed or in the process of deploying, its location is given by the following equations:

$$X_J = \phi_f \cos\theta - \beta \sin\theta$$

$$Y_J = \phi_f \sin\theta + \beta \cos\theta$$

where: $\phi_f = X_{J_f} \cos\theta + Y_{J_f} \sin\theta$

$$\beta = \frac{(-\sin\theta)(\sum_{i=1}^{N} X_{J_i}) + (\cos\theta)(\sum_{i=1}^{N} Y_{J_i})}{N}$$

10) Determining the deployment location for a member unit relative to the location of the operational grouping.

Given the following quantities:

a) The current location (IXYOG) of the operational grouping: (X,Y)

b) The direction of movement of the operational grouping: ($\sin\theta$, $\cos\theta$)

c) The member unit's planned (by input) location expressed in a coordinate system centered at the location of the operational grouping: (X^1,Y^1); X^1 is the forward distance from the current location of the operational grouping (positive in the direction of movement of the grouping), Y^1 is the lateral distance from the current location of the operational grouping (positive to the left and negative to the right when facing forward),

d) The expansion-contraction factor (FECFAC) for the operational grouping during deployment: F

The following equations give the coordinates of the deployment location (X_D, Y_D) for the member unit:

$$X_D = X + X^1 \cos\theta - (Y^1 \sin\theta)F$$

$$Y_D = Y + X^1 \sin\theta + (Y^1 \cos\theta)F$$

5.7 INPUT/OUTPUT MODULE

5.7.1 Input Submodule

5.7.1.1 Operation

The operation of the Input Submodule is simple and straightforward. It reads the data needed to run the model from the various data disk files using Fortran read statements, namelist INPUT statements and calls to the Fortran routines BUFFERIN and BUFFIN. The Input Submodule also packs data (puts more than 1 piece of data in 1 word of core) and initializes many of the arrays to zero. The significance of the Input Submodule is that it allows the CATTS model great flexibility since much of what the model does is controlled by the input data. This allowed, for example, the obsolete 90MM Recoilless Rifle to be replaced by the Dragon missile in the data base without changing a single line of the math model code. If the equipment, units, weapon effects, etc., were described in the code instead of in the data base, the replacement of one type of equipment with another would be much more difficult and expensive than it is in CATTS. In addition to equipment, a large amount of the tactically significant items are controlled and input via the data base. These include the relief, soil, vegetation, weather, and lighting conditions of the area of operations, the sensor types and placement, the units' TOE's and position, the weapon accuracy and effects data, tasking organization, modes of equipment operation, movement and change of state data, suppression data, and so on. Figure 5-100 shows the subroutine linkages for the Input Submodule, while Table 5-47 gives a brief description of each routine and its major inputs and outputs.

All the data needed to initialize and run the CATTS math model is on disk. Basically there are 9 data files needed to initialize and run the CATTS math model (there are other disk files needed to run associated with other modules - the background events file associated with the events module, for example).

The data files are:

1 terrain relief file

1 elevation displacement file
(needed only to pack the terrain relief data)

1 soil data file

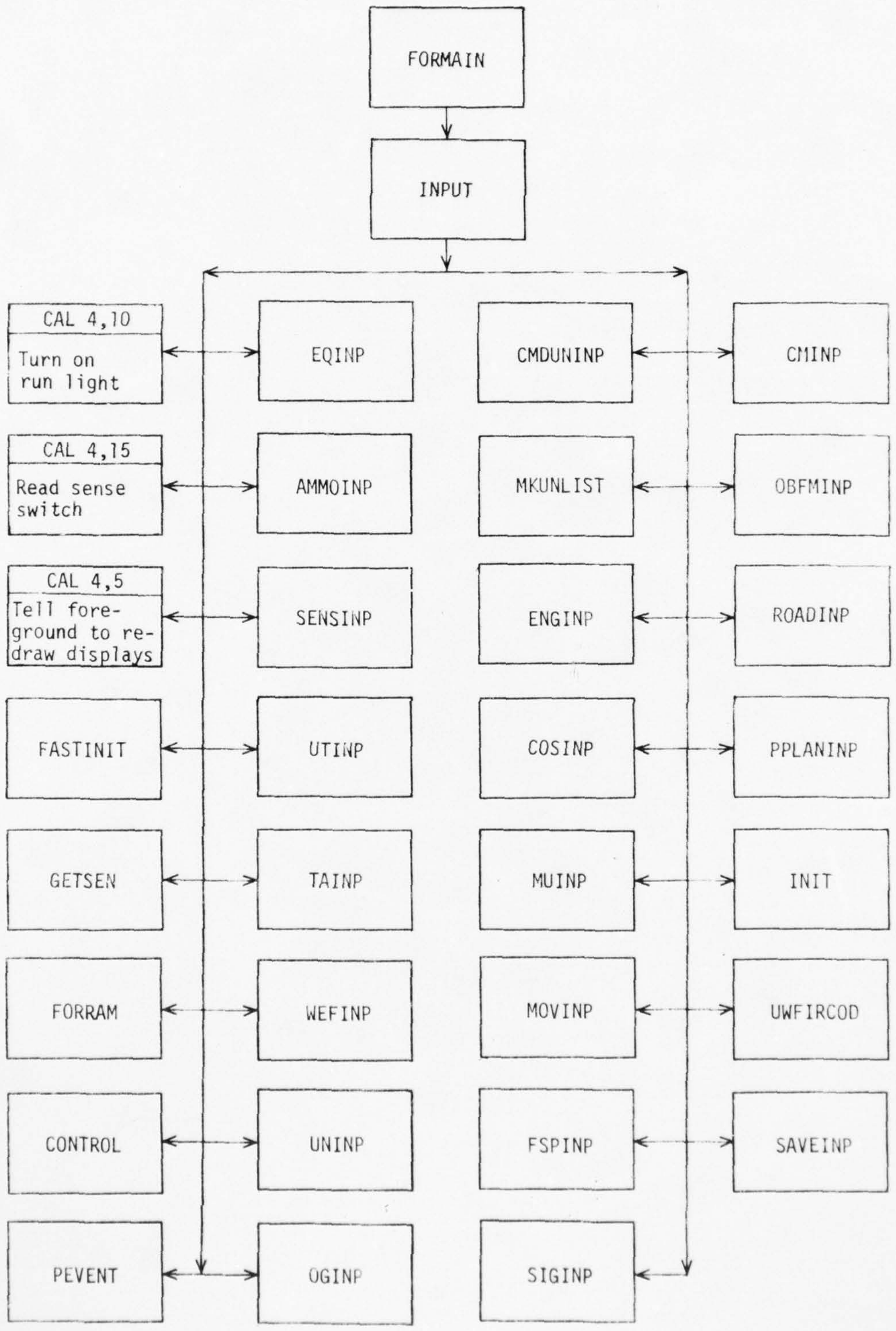


Figure 5-100. Subroutine Linkage of Input Submodule

Table 5-47. Input Submodule Subroutine Descriptions

• Subroutine	Description	Principal Inputs/Outputs		
INPUT	Reads basic inputs to the model from disk; number of different equipment types, time at which simulation run is to end, etc.	Input:	(See descriptions of Miscellaneous Variable Input Deck, Operational State Name Input Deck, Miscellaneous Input Deck, and Namelist Input Deck on pages 12-16, 97-100, 150-151, and 152-167 of Table 3-1 of the <u>Data Base/Operations Manual</u>.)	
		Output:	BLDIES(JU)	- Basic load (initial amount) of diesel fuel (in gallons) for unit JU ($1 \le JU \le 100$).
			BLGAS(JU)	- Basic load (initial amount) of gas (in gallons) for unit JU ($1 \le JU \le 100$).
			CLDIES(JU)	- Current load (present amount) of diesel fuel (in gallons) for unit JU ($1 \le JU \le 100$).
			CLGAS(JU)	- Current load (present amount) of gas (in gallons) for each unit JU ($1 \le JU \le 100$).
			ITMSTRT	- Time (in units of clock minutes) at which the simulation starts.
			NWORD1	- Number of words contained in first part of COMMON saves (root).
			NWORD2	- Number of words contained in second part of COMMON saves (root).
			NWORD3	- Number of words contained in third part of COMMON saves (segment one).

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
INPUT (Cont'd)		NWORD4 - Number of words contained in fourth part of COMMON saves (segment two). NWORD5 - Number of words contained in fifth part of COMMON saves (segment three). TSRX - Time of sunrise (decimal hours). TSSX - Time of sunset (decimal hours). VISM - Global meteorological visibility (in meters).
AMMOINP	Reads in the number of ammo types, ammo names, and which army the ammo can belong to. It also packs ammo ownership information into IAMOSIDE.	Input: (See description of Ammunition Input Deck on pages 26-28 of Table 3-1 of the Data Base/Operations Manual.) Output: IAMMONAM(M,IAMMO) - A 12 character (M = 1,3) name for each of the possible (IAMMO = 1,80) ammunition types. IAMOSIDE(IAMMO) - A byte-packed array (IAMMO = 1,20) where each byte contains a code controlling which types will appear on the menus when the red or blue command and control button is pushed. 0 = Both on red and blue 1 = Red only 2 = Blue only 3 = Neither red or blue (spare)

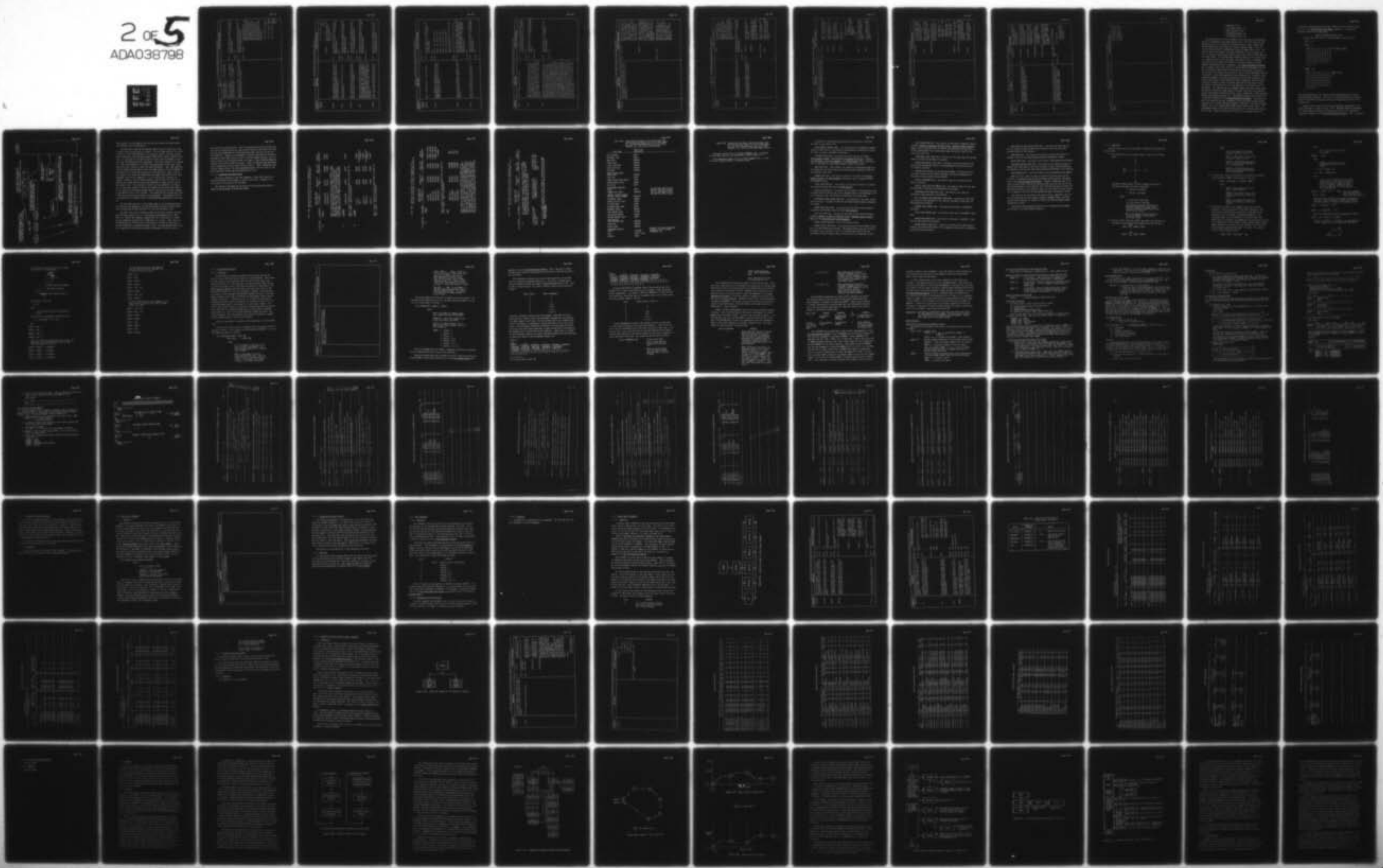
Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs	
AMMOINP (Cont'd)			NUMAMO - Number of the largest numbered ammunition type modeled.
CMDUNINP	Reads in the number of red and blue command units and the definitions of each (name and size).	Input:	(See description of Command Unit Input Deck on pages 85-91 of Table 3-1 of the Data Base/Operations Manual.)
		Output:	None
CMINP	Reads in the initial control measures; converts data to event format, and calls CONTMS to add them to data base.	Input:	(See description of Control Measure Input Deck on pages 122-128 of Table 3-1 of the Data Base/Operations Manual.)
		Output:	None
COSINP	Reads in lower and upper bounds for specially defined OP states, special threat states, backward moving states, the change of state table, and the decision value table.	Input:	(See description of Change of State Input Deck on pages 94-96 of Table 3-1 of the Data Base/Operations Manual.)
		Output:	None
ENGINP	Reads and sets up the initial engagements.	Input:	(See description of Engagement Input Deck on pages 92-93 of Table 3-1 of the Data Base/Operations Manual.)
		Output:	None
EQINP	Reads in number of equipments defined and, for each equipment, its name, which forces the equipment can belong to, weapon code, equipment target weight, number of persons required to operate equipment, number of personnel losses per equipment destroyed, maximum and minimum ranges, levels of noise produced by moving and stopped equipment, preferred primary and secondary target types, eight modes of operation with each mode characterized by the type of ammunition used, rate of movement, rate of fire,	Input:	(See description of Equipment Input Deck on pages 17-35 of Table 3-1 of the Data Base/Operations Manual.)
		Output:	None

AD-A038 798   TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF   F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER   N61339-73-C-0156

UNCLASSIFIED   NAVTRAEQUIPC-73-C-0156-E00   NL

2 OF 5
ADA038798



2 OF 5

ADA038798

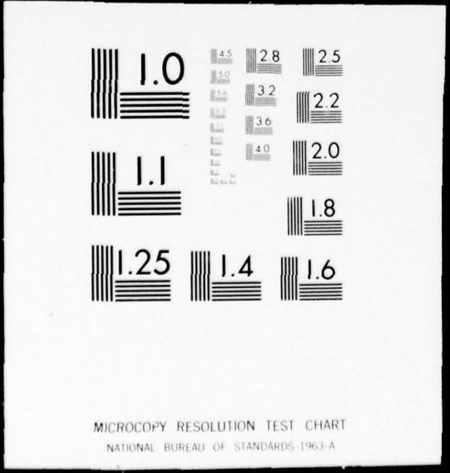


Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
EQINP (Cont'd)	and personnel vulnerability class, gasoline and diesel fuel consumption, and the fuel carrying capacity of the equipment.	
FSPINP	Reads in fire support safety ranges, lateral distance beyond half-width of engagement in which target may be assigned, support fire band definitions, and the fire support look-up table.	Input: (See description of Fire Support Input Deck on pages 111-118 of Table 3-1 of the Data Base/Operations Manual.) Output: None
MKUNLIST	Reads in units' next higher command chain and packs data into arrays so that battalions, companies, and platoons can be accessed independently.	Input: (See description of Higher and Adjacent Unit Input Deck on pages 87-91 of Table 3-1 of the Data Base/Operations Manual.) Output: LISTUN(IU) - A byte-packed array of pointers used to control the order which units are listed in on all the menus and the status report. Each word corresponds with a unit IU. The word points to the next unit IU in the list after IU using the following method: Each byte of the word contains the unit number of the next unit in the proper size list. ($1 \leq IU \leq 150$) 0 Byte (left-most)--Platoon list 1 Byte (left-middle)--Company list 2 Byte (left-middle)--Battalion and above list

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
MKUNLIST (Cont'd)		3 Byte (right-most)--List of all units NXHGCM(IU) - The unit number of the unit which is the next higher command for unit IU. (IU = 1,150)
MOVINP	Reads in definitions of special routes, the movement code change table, the operational state update table, the expansion and contraction table, and the movement data pool.	Input: (See description of Movement Inputs Deck on pages 105-110 of Table 3-1 of the Data Base/Operations Manual.) Output: None
MUINP	Reads in the mode selection table and the mode distribution table.	Input: (See description of Equipment Mode Input Deck on pages 100-104 of Table 3-1 of the Data Base/Operations Manual.) Output: None
OBFMINP	Reads in the definitions of obstacles including their name, type, side, number of segments and the width and location of the segments.	Input: (See description of Mine Obstacle Fortification Input Deck on pages 129-132 of Table 3-1 of the Data Base/Operations Manual.) Output: None
OGINP	Reads in the initial operational grouping definitions each containing a name, op group type, travel code, deployment code, size, op group movement code and movement data values, direction of movement, depth when engaged, frontage expansion-contraction factor, normal half-frontage, and minimum distance permitting transfer to new, closer engagements.	Input: (See description of Operational Grouping Input Deck on pages 81-84 of Table 3-1 of the Data Base/Operations Manual.) Output: None
PPLANINP	Reads in and stores preplanned mission names and event notices defining the preplanned missions.	Input: (See description of Preplanned Mission Input Deck on pages 145-149 of Table 3-1 of the Data Base/Operations Manual.)

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
PPLANINP (Cont'd)		Output: None
ROADINP	Reads in the definitions of roads and bridges.	Input: (See description of Road Input Deck on pages 133-141 of Table 3-1 of the <u>Data Base/ Operations Manual</u>.) Output: None
SAVEINP	Entry SAVEINP saves COMMON block groups 1-5 onto binary data base. Entry FASTINIT restores these COMMON block groups when initialization with the binary data base is desired.	Input: NWORD1 - (See subroutine INPUT.) NWORD2 - (See subroutine INPUT.) NWORD3 - (See subroutine INPUT.) NWORD4 - (See subroutine INPUT.) NWORD5 - (See subroutine INPUT.) (Under entry point FASTINIT, reads in binary input file (F:10).) Output: (Writes to disk, binary input file (F:10).)
SENSINP	Reads in various coefficients, parameters, and other data values defining various sensors used in the scenarios.	Input: (See descriptions of Sensor Input Deck, Visual Detection Input Deck, Aural Detection Input Deck, Unattended Ground Sensor Input Deck, and Ground Surveillance Radar Input Deck on pages 29-51 of Table 3-1 of the <u>Data Base/ Operations Manual</u>.) Output: None
SIGINP	Reads in the Suppression Criterion Selection Table and a set of four-point suppression curves.	Input: (See description of Suppression Input Deck on pages 119-121 of Table 3-1 of the <u>Data Base/ Operations Manual</u>.) Output: None

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
TAINP	Reads in target acquisition curves.	Input: (See description of Target Acquisition Input Deck on pages 58-59 of Table 3-1 of the Data Base/Operations Manual--Not used in CATTS.) Output: None
UNINP	Reads in the definitions of each unit including its name, op group, type of unit, initial op state, unit width, unit depth, initial travel code, unit size, movement data, initial location planned, location relative to op group, initial direction of movement, initial personnel mix, equipment types and amounts, ammo types and amounts, initial obstacle delay counter, and initial unit general delay counter.	Input: (See description of Unit Input Deck on pages 72-80 of Table 3-1 of the Data Base/Operations Manual.) Output: None
UTINP	UTINP reads in data values characterizing the various unit types. For each type of unit, these data include average unit height, width, and reflectance coefficient used for visual target acquisition, manning priorities for up to forty equipment types, data indicating which equipment categories are to be used to determine unit rate of movement, maximum distance forward of friendly units for use of indirect fire weapons, ranges at which unit is eligible to initiate engagement with units of other types, ranges at which unit must initiate an engagement, weight factor of unit as support fire target, maximum distance beyond the end of the enemy FEBA in an established engagement that an unengaged unit of this unit type will be allowed to join an existing engagement, rate of air speed if airborne unit, weight factor expressing the increase in importance as direct, indirect, and support fire targets when unit is in one of the special threat states, and maximum ranges at which it can employ direct, indirect, and support fire against enemy targets.	Input: (See description of Unit Type Deck on pages 52-57 of Table 3-1 of the Data Base/Operations Manual.) Output: None

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
UWFIRCOD	Packs the ground fire module debug print NAMELIST input flags in bit array IUCOD and IWCOD. Since a temporary array is used, the debug print option is reset every run.	Input:	IUNTFIR(K)	- Number of K-th unit for which a fire report print is requested ($1 \leq K \leq 10$). A value of zero terminates the list of units. A value greater than 100 requests the report for all units. This table is used for input only. For internal model use, the table IUCOD is constructed and stored.
			IWPNFIR(K)	- Number of the K-th weapon type for which a fire report print is requested ($1 \leq K \leq 10$). A value of zero terminates the list. A value greater 80 requests the report for all weapons. This table is used for input only. For internal model use, the table IWCOD is constructed and stored.
		Output:	IUCOD	- Bit-packed version of IUNTFIR. Each of the first 99 bits of IUCOD corresponds to a single CATTS unit (bit one to unit one, etc.). A zero in the bit position requests no fire report for the corresponding unit, a one requests the fire report for the unit.

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
UWFIRCOD (Cont'd)			IWCOD	- Bit-packed version of IWPNFIR. Each of the first 80 bits of IWCOD corresponds to a single CATTS equipment type (bit one to equipment type one, etc.). A zero in a position requests no fire report for the corresponding equipment type, a one requests the fire report for that equipment type.
WEFINP	Reads in the Weapon Effects Look-up Table, a set of weapon effect coefficients and special coefficients for weapon effect function 6.	Input:	(See description of Weapon Effects Input Deck on pages 60-71 of Table 3-1 of the Data Base/Operations Manual.)	
		Output:	None	
CONTMS	Processes interactive command and control events to add or delete control measures. Also used to process control measures input via the data base.	Input:	INVDT(JEVT)	- Event notice data of event that is being processed. ($1 \le JEVT \le 64$)
			MTYPOFCM(I)	- A byte-packed array. Byte J gives the background index for the control measure type which has foreground index J. ($1 \le I \le 35$)
		Output:	ICM(J,ICM)	- Definition of ICM-th control measure. ($1 \le ICM \le 100$)
			NOCMT	- Highest value of I for which ICM(1,I) is non-zero; that is number of control measures in table.

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
INIT	Initializes various counters and storage locations needed during the course of the run. Also, certain inputs required for old MAFIA restarts are read by INIT.	Input: None Output: DNU(JU,J) - Casualties for equipment J in unit JU during current time-step. ($1 \le JU \le 100$, $1 \le J \le 14$) DPERS(JU) - Personnel casualties for unit JU during current time-step. ($1 \le JU \le 100$) DPERSU(JU) - Casualties for unit JU if unit JU is unsuppressed during current time-step. ($1 \le JU \le 100$) ISTATU(JU) - Status code of unit JU, ($1 \le JU \le 100$): = -1 Air unit or defunct ground unit = 1 in engagement, eligible for direct fire against enemy units in same engagement = 0 Other IWPTS - Used to save weapon number from one weapon table search to another by subroutine SCHRMU. IWPTST - User to save weapon number from one weapon table search to another by subroutine WPNFIR.

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
INIT (Cont'd)		MUSLRA(JU) - Distance (in meters) of unit JU to enemy FEBA or nearest enemy unit; used to select proper mode distribution vector. ($1 \le JU \le 100$) NPFP1 - NPFC + 1 where NPFC is the number of entries in weapon effects table involving personnel targets. PERNVC(JU,IPVC,K) - Number of personnel in each vulnerability class IPVC, ($1 \le IPVC \le 6$), for unit JU, ($1 \le JU \le 100$), where K: 1 = Actual number in previous time-step 2 = Actual number in current time-step 3 = Unsuppressed number in previous time-step 4 = Unsuppressed number in current time-step PERSWT(JU) - Weight of unit JU as a personnel target. ($1 \le JU \le 100$) STATS(JEQ,J,JCOLOR) - Equipment and personnel casualty statistics for red killing blue (JCOLOR=1), and blue killing red (JCOLOR=2) equipments. SWFU(JU) - Area support fire weapon rounds/unit time received by unit JU. ($1 \le JU \le 100$)

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
INIT (Cont'd)		USEEQU(JU,J) - Number of pieces manned for the J-th equipment type carried by unit JU. ($1\leq JU\leq 100$, $1\leq J\leq 14$)
MINEFLDS	Calculates the four X, Y coordinate points of the corners of the rectangles which are used to represent minefields. Uses the data base input line segment from the obstacle input deck.	Input: IOBX(I,IOBS) - X-coordinate of end point of the I-th, ($1\leq I\leq 6$), line segment describing the IOBS-th, ($1\leq IOBS\leq 50$), obstacle. IOBY(I,IOBS) - Y-coordinate of end point of the I-th, ($1\leq I\leq 6$), line segment describing the IOBS-th, ($1\leq IOBS\leq 50$), obstacle. MINEDATA(JMNFLD) - Half-word packed array containing data pertaining to the JMNFLD-th, ($1\leq JMNFLD \leq 20$), minefield. MNEFLDXY(I,J,JMNFLD) - Half-word packed array containing the X and Y coordinates of the end points of the line segments describing the JMNFLD-th, ($1\leq JMNFLD\leq 20$) minefield.
TESTF10	Checks the record size (RSIZE) parameter of the RAD File Table for F:10 to ensure that the RSIZE IS large enough to hold the common blocks (4 times the largest block to be written - defined by NWORD1, NWORD2, NWORD3, NWORD4, and NWORD5).	Input: (Looks at DCB - see Xerox RBM Reference Manual - for F:10.) Output: IND - Flag indicating: = -1 Delete events of type IT1 and subtype IT2 occuring prior to or at time ITM

Table 5-47. Input Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
TESTF10 (Cont'd)		= 0 Delete events of type IT1 and subtype IT2 occuring at time ITM = 1 Delete events of type IT1 and subtype IT2 occuring at or after time ITM

2 vegetation files
1 card image scenario file
1 binary scenario file
1 prescheduled events file
1 pre-planned mission file

One additional data file, not needed to run, but used for configuration control is the baseline card image scenario file. This file serves as the beginning point for all changes to the scenario but is never changed until all the changes have been tested and used for running a game. This file contains the same type of information as the card image scenario file. The binary scenario file contains the same information as the card image file, but in a different format which allows a much quicker initialization than the card image format. All the detailed information about all of the data files i.e., where the files are on the disks, what specific data is contained in each file, what the format of the data is, and which scalars and arrays (tables) the data goes into is contained in the Data Base/Operations Manual. But, in general, the terrain relief file and elevation displacement file contain the elevation data (some 4 million points) for the area of operation. The soil data file contains the location of the soil types throughout the area of operation. One of the vegetation files contains the X,Y locations of the vegetation polygons in the area of operations and the other vegetation file contains descriptions of the vegetation classes themselves (the description of the soil classes is input via the UGS input deck on the scenario file). The scenario file contains all tactical data that is not terrain, soil, or vegetation. This includes data on equipment, ammunition types, sensor types, UGS types and locations, radar types, unit types, unit TOE's and locations, tasking organization, minefield location, roads, suppression, control measures, weather, pre-planned missions, etc. The method of updating the scenario file is detailed in Section 4 of the Data Base/Operations Manual. The binary scenario file contains the same information as the card image scenario except the data is in 5 records each approximately 16,000 words long (the card image scenario files are in approximately 4000 records each 20 words long). The prescheduled events file contains events which will occur every time the scenario is run. These events are put on this file via punched cards or an initialization procedure that puts events input during initialization onto this file - see Data Base/Operations Manual, Appendix B. A sample prescheduled events file is shown below.

Sample Prescheduled Events File
(with one fire event (Type 10) and one maneuver event (Type 9))

```
   NEVTP = 2,
*
  ITMEVE = 0,
    IVDT = 10, 1, 25, 17, 2, 1, 201, 6, 64750, 19900,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 2,
*
  ITMEVE = 15,
    IVDT = 9, 10, 24, 3, 0, 0, 1, 55600, 21200,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, C, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 2,
*
```

The pre-planned mission file contains the pre-planned missions to be executed during the simulation. This file is filled with data by the computer during initialization using data from the pre-planned mission deck on the scenario file.

The data flow from the disk files to the computer core memory for use by the math model is shown in Figure 5-101. The exact checklist procedures for initializing the simulation including sense switch 3 and ISAVEINP are contained in Appendix B of the Data Base/Operations Manual. Thus, the follow-

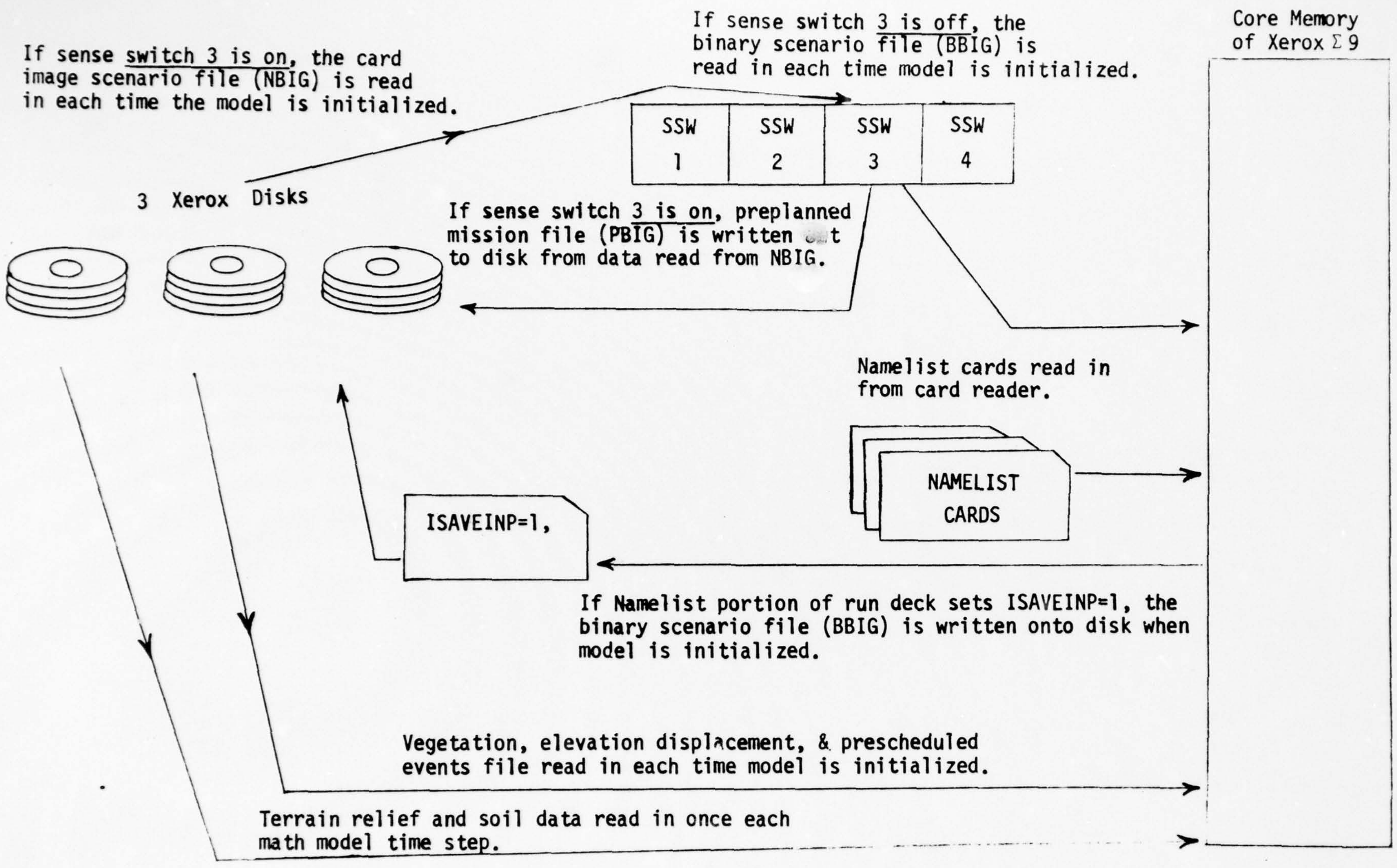

If sense switch 3 is on, the card image scenario file (NBIG) is read in each time the model is initialized.
3 Xerox Disks
If sense switch 3 is off, the binary scenario file (BBIG) is read in each time model is initialized.
Core Memory of Xerox Σ 9
SSW 1
SSW 2
SSW 3
SSW 4
If sense switch 3 is on, preplanned mission file (PBIG) is written out to disk from data read from NBIG.
Namelist cards read in from card reader.
NAMELIST CARDS
ISAVEINP=1,
If Namelist portion of run deck sets ISAVEINP=1, the binary scenario file (BBIG) is written onto disk when model is initialized.
Vegetation, elevation displacement, & prescheduled events file read in each time model is initialized.
Terrain relief and soil data read in once each math model time step.

Figure 5-101. Data Base Information Flow (Example For Scenario BIG)

ing discussion is an attempt to give the user the reasons and theory behind the different files and procedures.

The data base is kept on disk because tape or cards are both much slower to read in and are much bulkier to handle. The card image scenario file looks like cards, in fact (80 characters or 20 words per record), and takes approximately 5 minutes to read the approximately 4000 records (cards) comprising this file. If the scenario file were actually on cards, the time needed to input this same information would be even greater. The binary scenario file, because it is read in by BUFFERIN in 5 records each approximately 16,000 words long, takes less than 1 second to input this same information. Thus, SSW3 should be on when changes have been made in the card image scenario file and ISAVEINP is set equal to 1 to insure the creation of a new binary file. Also, a new binary scenario file should be created when any common block definition or data statement in the root or segments 1, 2 or 3 of the overlay structure are changed. Normally, only for testing purposes would a user turn SSW3 on to read the card image scenario and not set ISAVEINP to create a binary copy of the scenario file. Also note that if SSW3 is off but ISAVEINP is set equal to 1, a new binary scenario file will be created from the old binary scenario file with the only difference between new and old being the variables changed by the namelist cards in the run deck. This capability has not been used often in the past, but could be a method of testing changes. The problem with this is the difficulty in keeping track of and documenting changes made in this manner.

The only other data file in card image format is the prescheduled events file, and this was designed so the user could punch up cards in event notice form and copy them directly onto this file and have the events occur.

The other files are all blocked so they can be read in quickly, the same way the binary scenario file is. The elevation displacement file is 2250 words in 1 record, the vegetation composition file is 265 words in 1 record and the vegetation location file is 2828 words in 1 record. These files are not packed but are in binary form so that they can be read in, and with no further translation or packing are ready to use. The prescheduled events file, the 2 vegetation files and the elevation displacement file are all read in only once during initialization. The file containing the terrain relief data and the file containing the soil data are both read once each model time step. This has the disadvantage of slowing down each model time step to read this data from disk. However, this procedure is necessary because the information contained on these two files will not fit into core memory and thus, must be read in pieces using the double buffers (as shown in Table 5-48) to speed the reading of this data as much as possible. Table 5-48 details how the terrain relief, vegetation, and soil data files are read in. Table 5-49 details how the card image scenario file and prescheduled events files are read in, and which subroutine creates the pre-planned mission file.

5.7.1.2 Assumptions and Data Sources

The basic assumption of the input submodule is that CATTS should be a data driven model so that changes in tactics, unit TOE's, equipment, etc. can be easily incorporated into the model.

The sources of the data for the various files are many and varied, as shown in the paragraphs following the tables.

Table 5-48. Subroutines that Read in Relief, Vegetation, and Soil Data into Arrays for Use by the Math Model

F: # of File	File Name	Array Name That Data From File is used in	Common Block Name	Read in by Subroutine	Used in Subroutine
18	RELIEF				
		IELEVE(34,472)	RELIEFE	LOSCOMP	LOSVEG
		IELEVO(34,472)	RELIEFO	LOSCOMP	LOSVEG
	Data is read in each minute using 2 (double) buffers. Data read in one buffer and next the other buffer (first IELEVO, then IELEVE, then IELEVO, etc.) to do calculation while reading next block of data into buffer not being used in calculations. Flip-flops depend on variable INDELV. Only blocks needed are read in. Read in by BUFFIN and BUFFERIN.				
21	ELEVDISP				
		IEVDSP(3,750)	ELVDSP	LOSINP	LOSVEG
	Data is read in initially to fill common block and then not read in again. Read in by BUFFERIN.				
22	VEGCOMP				
		H(16,4)	DATAB1	LOSINP	LOSVEG
		HB(9)	DATAB1	LOSINP	No longer used in CATTS
		RHO(16,4)	DATAB1	LOSINP	LOSVEG
		RMAX(16,4)	DATAB1	LOSINP	No longer used in CATTS
		W(16,4)	DATAB1	LOSINP	LOSVEG
	Data is read in initially to fill common blocks and then not read in again. Read in by BUFFERIN.				

Table 5-48. Subroutines that Read in Relief, Vegetation, and Soil Data into Arrays for Use by the Math Model (Cont'd)

F: # of File	File Name	Array Name That Data From File is used in	Common Block Name	Read in by Subroutine	Used in Subroutine
23	VEGLOC				
		ICL(225)	DATAVEG	LOSINP	LINOBS & MICSOL
		ITRC(225)	DATAVEG	LOSINP	LINOBS & MICSOL
		XPOLY(5,225)	DATAVEG	LOSINP	LINOBS & MICSOL
		YPOLY(5,225)	DATAVEG	LOSINP	LINOBS & MICSOL
		IVEGLOC(2,64)	DATAVEG	LOSINP	LOSCOMP

Data is read in initially to fill common blocks and then not read in again. Read in by BUFFERIN.

F: # of File	File Name	Array Name That Data From File is used in	Common Block Name	Read in by Subroutine	Used in Subroutine
24	SOIL				
		ISOILE(361)	RELIEFE	LOSCOMP	SOIL
		JSOILE(15687)	RELIEFE	LOSCOMP	SOIL
		ISOILO(361)	RELIEFO	LOSCOMP	SOIL
		JSOILO(15687)	RELIEFO	LOSCOMP	SOIL

Data is read in each minute. After the last block of the RELIEF file is read in and while the calculations are being made on it, the soil data is read into either the IELEVE array or the IELEVO array. When subroutine LOSCOMP finishes the last LOS calculation, it returns to entry LOSINP1 which then immediately calls subroutine SOIL. The arrays in subroutine SOIL associated with common block RELIEFE or RELIEFO now contain the soil data from disk file SOIL even through the data was read in by LOSCOMP using arrays IELEVE or IELEVO.

Table 5-48. Subroutines that Read in Relief, Vegetation, and Soil Data into Arrays for Use by the Math Model (Cont'd)

F: # of File	File Name	Array Name That Data From File is used in	Common Block Name	Read in by Subroutine	Used in Subroutine
24	This is because data is stored in core in the area allocated by common block name. The variable INDELV, which is in common block ELVDSP indicates to subroutine SOIL whether the data is in arrays ISOILE and JSOILE or ISOILO and JSOILO.				

In subroutine LOSCOMP array IELEVE is in this area.

CORE AREA RESERVED FOR COMMON BLOCK RELIEFE

In subroutine SOIL, arrays ISOILE and JSOILE are in this area.

Note- F: # is a number assigned via Xerox !ASSIGN or :ASSIGN Command (see RBM Reference Manual) that tells the math model where on disk a particular file is located.

Table 5-49. Subroutines that Read in the Scenario Data from the Data Base Card Image Files (F:5) NBIG, NGOLD, etc., and Prescheduled Events File (F:11) EBIG, EFBAGOLD, etc., for Use by the Math Model

Deck Name	Read in by Subroutine	
MISC VARIABLE	INPUT	
EQUIPMENT INPUT	EQINP	
AMMO INPUT	AMMOINP	
SENSOR INPUT	SENSINP	
VISUAL DETECTION	SENSINP	
AURAL DETECTION	SENSINP	
UGS INPUT	SENSINP	
GROUND SURVEILLANCE RADAR INPUT	SENSINP	
UNIT TYPE	UTINP	
TARGET ACQUISITION INPUTS	TAINP	
WEAPON EFFECTS INPUT	WEFINP	
UNIT INPUT	UNINP	
OPERATIONAL GROUPINGS INPUTS	OGINP	All the data shown on this table is read in initially to fill the common blocks and then not read in again.
COMMAND UNIT INPUT	CMDUNINP	
CONTROL LISTING OF UNITS AND NEXT HIGHER COMMAND	MKUNLIST	
ENGAGEMENT INPUT	ENGINP	
CHANGE OF STATE INPUT	COSINP	
OP STATE NAME	INPUT	
EQUIPMENT MODE INPUT	MUINP	
MOVEMENT INPUTS	MOVINP	
FIRE SUPPORT INPUT	FSPINP	
SUPPRESSION INPUTS	SIGINP	
CONTROL MEASURE INPUT	CMINP	
MINE OBSTACLE FORTIFICATION INPUT	OBFMINP	
ROAD INPUT	ROADINP	
BRIDGE INPUT	ROADINP	
PREPLANNED MISSION INPUT	PPLANINP	Outputs info onto preplanned mission file(F:12) PBIG, PFBAGOLD, etc.
MISC	INPUT & INIT	
NAMELIST	INPUT	

Table 5-49. Subroutines that Read in the Scenario Data from the Data Base Card Image Files (F:5) NBIG, NGOLD, etc., and Prescheduled Events File (F:11) EBIG, EFBAGOLD, etc., for Use by the Math Model (Cont'd)

The binary scenario file (F:10) BBIG, BFBAGOLD, etc., is created using the data read in by the subroutines listed in Table 5.7-2.

The prescheduled events file (F:11) EBIG, EFBAGOLD, etc., is read in by local subroutine GETEV in subroutine INPUT.

Following is a deck by deck description of the sources of the CATTS data base on the scenario file.

MISCELLANEOUS VARIABLE DECK. This deck consists of bookkeeping numbers, except for WPVC, which is the weight of the personnel vulnerability classes. This data is based on the MAFIA V Data Base.

EQUIPMENT INPUT DECK. The sources for this data are as follows: Jane's Weapons Systems, Staff Officer Field Manual FM 101-10-1, Tactical Vehicles Manual, Artillery Handbook, the STANO Handbook, MAFIA V Data Base, Dash 1 Flight Manuals, and USAREUR Pamphlet 30-60-11 and 30-60-1.

AMMO INPUT DECK. These numbers are essentially bookkeeping numbers to give ammunition the proper name on the menus and were produced by TRW from the equipment input deck.

SENSOR INPUT DECK. The sources of data for this deck include Jane's Weapons Systems, the STANO Handbook, Biocoustics Labs at Wright-Patterson and Edgewood Arsenal.

VISUAL DETECTION DECK. The sources for this data include Iran Counter-Infiltration Systems Study and the STANO Handbook.

AURAL DETECTION DECK. The sources for this data is the Biocoustics Labs at Wright-Patterson AFB and the lab at Edgewood Arsenal, the Audiology Clinic at Fort Benning, SIAF, and Technical Discussions with Captain Amos.

UNATTENDED GROUND SENSOR INPUT DECK. The sources for this data include Terrain Analysis in Support of CATTS by the Defense Mapping Agency Topographical Center.

GROUND RADAR INPUT DECK. The sources for this data include Technical Discussions with Reinhart Olesch and the STANO Handbook.

UNIT TYPE INPUT DECK. Sources for this data include the GFP Scenario Package, Handbook on Aggressor Military Forces, Aggressor Order of Battle Book, and the Operations and Training Handbook.

TARGET ACQUISITION DECK. This is not used by CATTS.

WEAPON EFFECTS INPUT DECK. This data was derived from the MAFIA V Data Base, and TRW Engineering Estimates. ADA weapon effects data was furnished by Captain Larry Lippincott, and air to ground weapon effects data was furnished by Colonel Gardner, both in discussions at Fort Benning in 1975.

UNIT DESCRIPTION DECK. Data sources for this deck include GFP Scenario Package, Handbook on Aggressor Military Forces, Aggressor Order of Battle Book, Infantry Reference Data, GFP Tables of Organization and Equipment (H Series), Memorandums from Colonel McGilicuddy and Colonel Franklin, and Type Corps Troop List.

OPERATIONAL GROUP INPUT DECK. The data for this deck came from the GFP Scenario Package and the MAFIA V Data Base.

HIGHER AND ADJACENT INPUT DECK. The data in this deck came from the GFP Scenario Package and several of the memorandums and discussions with Fort Benning personnel.

CONTROL DECK FOR THE LIST OF UNITS ON THE MENU. The source data for this deck includes the GFP Scenario Package, and several memoranda and discussions with personnel at Fort Benning.

INITIAL ENGAGEMENT DECK. The source of data for this deck was the MAFIA V Data Base.

UNIT OP STATE DECISION CONTROL DECK. The source of data for this deck was the MAFIA V Data Base and TRW Engineering Estimates.

OPERATIONAL STATE NAME INPUT DECK. The source of this data was discussions with personnel at Fort Benning.

DECK TO CONTROL HOW OPERATIONAL STATES WORK. The source of this data is discussions with personnel at Fort Benning, the MAFIA V Data Base, and TRW Engineering Estimates.

MOVEMENT CODE CONTROL DECK. The source of this data is the MAFIA V Data Base.

FIRE SUPPORT CONTROL DECK. The source of this data is the MAFIA V Data Base.

SUPPRESSION CONTROL DECK. The source of this data is the MAFIA V Data Base and TRW Engineering Estimates.

CONTROL MEASURE INPUT DECK. Source of this deck was the GFP Scenario Package, and later modifications of this package by the user at Fort Benning during the evaluation phase.

MINE OBSTACLE FORTIFICATION INPUT DECK. The source of this data was the GFP Scenario Package and modifications made by the user during the evaluation phase at Fort Benning.

ROAD INPUT DECK. The source of this data is the Terrain Analysis in Support of CATTS by the Defense Mapping Agency Topographical Center.

BRIDGE INPUT DECK. The source of this data is TRW Engineering Estimates.

PREPLANNED MISSION INPUT DECK. The source of this data was letters of changes to be made dated 17 June 1975; 18 June, 19 June, CATTS Directorate, Fort Benning, and also user input during the evaluation phase at Fort Benning.

MISCELLANEOUS DECK. This deck contains solely bookkeeping type numbers.

NAME LIST DECK. Name list data at the end of the data base consists primarily of bookkeeping type values and debug control printout (see Appendix B of the Data Base/Operations Manual). However, EQOVRD and some of the other variables that relate to the obstacle model came from the user during the evaluation phase at Fort Benning, SIAF, Planning and Design of Roads, Air Bases and Heliports in the Theater of Operation, TM 5-330, and Terrain Analysis in Support of CATTS by the Defense Mapping Agency Topographical Center.

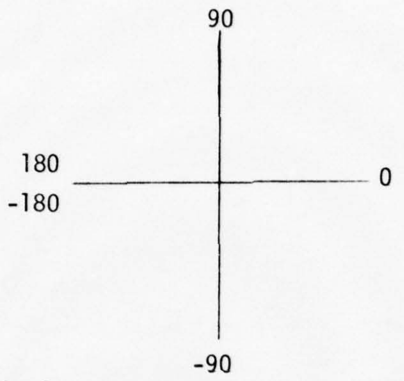
The source of all the data on the terrain relief, vegetation, and soil files in the CATTS data base (RELIEF, ELVDISP, VEGCOMP, VEGLOC, and SOIL) was the Terrain Analysis in Support of CATTS by the Defense Mapping Agency Topographical Center.

The data on the prescheduled events file is constantly being changed by the user to fit the different scenarios.

5.7.1.3 Equations

Obviously there are not a large number of equations associated with inputing data.

1) The direction a unit or op group faces is input as a cartesian angle



and the following formula is used to convert this value to sine and cosine which are used in the model.

$$S = SIN(DIR \cdot 0.0174533)$$

$$C = COS(DIR \cdot 0.0174533)$$

where

S is the sine of the angle.

C is the cosine of the angle.

SIN & COS are functions from the Xerox extended FORTRAN library that return the sine or cosine, respectively of any number they are given.

DIR is the input direction the unit is to face in degrees in the cartesian coordinate system.

2) The basic load of gasoline (BLGAS) and diesel fuel (BLDIES) are calculated using data from the equipment deck and unit deck.

$$BLGAS = \sum_{1}^{NETU} GCAPAC \cdot EQINIT$$

$$BLDIES = \sum_{1}^{NETU} DCAPAC \cdot EQINIT$$

where

> NETU is the number of different equipment types in a unit (1-14).
>
> EQINIT is the amount of each of the one to NETU types in a unit.
>
> GCAPAC is the gasoline capacity for each of the one to NETU equipments in a unit in gallons.
>
> DCAPAC is the diesel capacity for each of the one to NETU equipments in a unit in gallons.

3) The input days, hours, and minutes are converted to a standard time in minutes (JTIMEO) since midnight on day zero.

$$\text{JTIMEO} = 1440 \cdot \text{NDAYE} + 60 \cdot \text{NHOURE} + \text{NMINE}$$

where

> NDAYE is the calendar day of the start of the simulation.
>
> NHOURE is the number of hours since midnight at the start of the simulation.
>
> NMINE is the number of minutes into the hour at the start of the simulation.

4) Minefields are input as line segments along with all other obstacles in the mine obstacle fortification input deck. However, during initialization, minefields are converted to rectangles for use in the model in subroutine MINEFLDS. This subroutine takes the input endpoints' coordinates (OBX, OBY, OBX1 & OBY1) of the input line segment and calculates the four pairs of X,Y coordinates (MINEX1, MINEY1, MINEX2, MINEY2, MINEX3, MINEY3, MINEX4, MINEY4) of the corners of the rectangle, which are used by the obstacle model when the model is running. This conversion uses the following formulas.

$$\text{SLOPE} = (\text{OBY1} - \text{OBY})/(\text{OBX1} - \text{OBX})$$

where

SLOPE is the slope of the input line segment.

$$PERPEND = -1/SLOPE$$

where

PERPEND is the slope of a line perpendicular to the input line segment.

$$C = OBY - PERPEND \cdot OBX$$

$$C1 = OBY1 - PERPEND \cdot OBX1$$

where

C and C1 are the intercepts of the lines perpendicular to the input line segment, passing through each endpoint of the input line segment. These equations are the slope intercept form of an equation for a straight line.

$$COSEC\ A = \pm(COT^2 A + 1)^{1/2}$$

$$COT\ A = 1/TAN\ A$$

Note: input line segments are processed so that COSEC A is always positive.

where the above 2 formulas are standard trigonometric relations. Combining these 2 formulas with the fact that the slope of a line equals the tangent, we get:

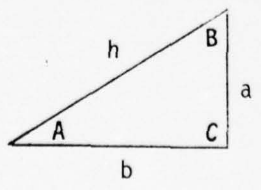
$$COSEC\ A = (1/(TAN\ A)^2 + 1)^{1/2}$$

$$COSEC\ A = (1/(PERPEND)^2 + 1)^{1/2}$$

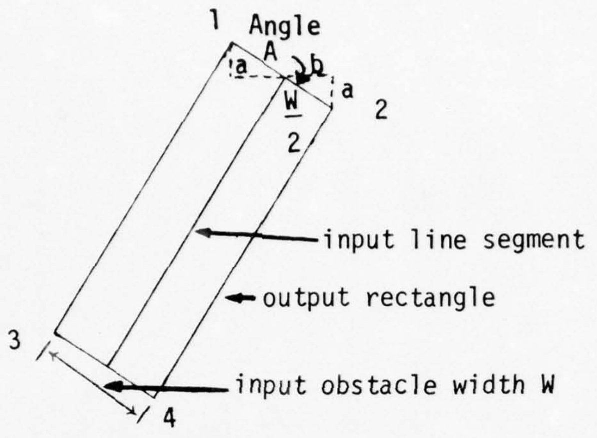
Next, the right-angle triangle trigonometric function

$COSEC\ A = h/a$ yields, $a = h/COSEC\ A$

where a is a side of the triangle, h is the hypotenuse of the triangle and A is the angle opposite side a.



The following illustration will aid in understanding the final calculations.



Noting that in this case

$h = W/2$

where

W is the input width of the obstacle in meters.

from the above trigonometric formulas we find

$a = (1/(\mathrm{PERPEND})^2 + 1)^{1/2} \cdot W/2$

and

MINEY1 = OBY + a

MINEY2 = OBY - a

MINEY3 = OBY1 + a

MINEY4 = OBY1 - a

The X's are then calculated using the Y values, the previously calculated intercepts, and the slope intercept equation of a line.

MINEX1 = (MINEY1 - C)/PERPEND

MINEX2 = (MINEY2 - C)/PERPEND

MINEX3 = (MINEY3 - C1)/PERPEND

MINEX4 = (MINEY4 - C1)/PERPEND

for cases where the input line segment is vertical and the slope is undefined, the following formulas are used:

MINEY1 = OBY

MINEY2 = OBY1

MINEY3 = MINEY1

MINEY4 = MINEY2

MINEX1 = OBX - W/2

MINEX2 = MINEX1

MINEX3 = OBX + W/2

MINEX4 = MINEX3

for cases where the input line segment is horizontal, and the slope is zero, the following formulas are used:

MINEY1 = OBY + W/2

MINEY2 = OBY - W/2

MINEY3 = MINEY1

MINEY4 = MINEY2

MINEX1 = OBX

MINEX2 = MINEX1

MINEX3 = OBX1

MINEX4 = MINEX3

5.7.2 Alert Message Submodule

5.7.2.1 Operation

The Alert Message Submodule generates alert messages when called upon to do so by the other various modules of the math model. These alerts are then passed to the foreground, which can then display them on the Super Bee terminals for the instructors to read and process (what the controllers may do with alerts at the Super Bee terminal is described in the CATTS Operator's Manual) and write them on the disk to be saved for the alerts post processor of the Alerts Submodule. However, since there are no routines in this submodule (ALERTGEN and ALNUALRT are part of the foreground software) Table 5-50 is presented for completeness rather than for information content. A complete list of all of the alerts and the subroutines where these alerts are generated is contained in Section 5.7.2.3 of the Programming Report and Section 3.11 of this User's Manual. The alerts cover a wide variety of situtations including beginning and ceasing fire, receiving fire, detecting enemy units, crossing control measures, firing short of or over firing control measures, damage and losses, personnel casualties, fuel and ammunition low, obstacle encounters, resupply actions, weather changes, and simulation control.

There are two methods of getting an alert generated in the math model code.

The first will allow the user to generate any of the alerts on disk in the alert library. This is done by calling subroutine ALERTGEN. A call to ALERTGEN would appear as follows:

```
CALL ALERTGEN(NT, NUNIT, ITEM1, NW1,
              ITEM2, NW2,. . ., ITEMK, NWK)
```

where

NT is the number of the type of the alert (1-120), (see Appendix C of the Programming report for a complete list of types).

NUNIT is the number of the unit associated with the alert. Zero is input for alerts not associated with a unit. This controls the routing of alerts. (See IUNIT discussion).

Table 5-50. Alert Message Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
ALERTGEN	(This is a foreground software subroutine--see Section 4.2.1.2.17 in the _Trainer Programming Report_ (pages 4-116 thru 4-124) for a description of this subroutine.)	

ITEM1, ITEM2,. . . ITEM K - These are the variable parts of the alerts, and they vary in number (1-K) and content depending on the alert type. They include things like unit names, locations, time, etc. They are specified as the first element of an array, a scaler, or Hollerith character strings.

NW1, NW2,. . . NWK - is the number of computer words associated with each of the same numbered items. For example, a 12 character unit name occupies 3 words of core.

The second method allows the user to generate any alert desired. The message must be loaded into an array, and then subroutine ALNUALRT called. This call would appear as follows:

```
CALL ALNUALRT(KNT, IARRAY(1), IROUTE)
```

where

KNT is the number of computer words that contain the alert (35 maximum).

IARRAY(1) is the first element of the array that contains the alert.

IROUTE is a number between 1 and 7 that controls which consoles the alert will be sent to:

IROUTE = 1	console	1
2	console	2
3	console	3
4	console	1 & 3
5	console	2 & 3
6	console	1 & 2
7	console	1, 2, & 3

Thus, by encoding time, unit names, or whatever is desired, any message may be constructed and sent to any of the controllers.

New alert message types may be added to the alert library by running program ALIBPROG as described in Section 4.4.15 of the Programming Report and

Appendix B of the Data Base/Operations Manual. Once a new type is added, ALERTGEN can be called by the appropriate subroutine to send the new type to the controllers.

Any alphanumeric message can be sent to the controllers at any model time by using the prescheduled events file. To do this, an event type 5 should be punched on cards and added to the prescheduled events file[1]. Word 1(IVDT(1)) should be equal to 5, word 2 (IVDT(2)) should be the routing as follows:

IVDT(2) VALUE	ROUTE TO CONSOLE #
1	1
2	2
3	3
4	1 & 3
5	2 & 3
6	1 & 2
7	1, 2, & 3

and words 3 through 37 should contain the message. Note that since the event must be in namelist form, the alphanumeric message must be converted to integer numbers to be input. This can be done easily by a program which reads in the message in alphanumeric format and writes it out using integer format. Such a program (called A2I and stored in the Data Base card file) was used by the field service personnel to punch up the alerts that generate the messages which identify the various scenarios. Shown below is a sample of how the message identifying scenario FEBA GOLD appears after being punched on cards, and copied to the prescheduled events file.

```
 NEVTP= 2,
*
 ITMEVE=          0,
 IVDT=5 ,7 ,-472259008 ,-488054045 ,-975904192 ,-942024988 ,-680836927 ,
 1088866518 ,-741087168 ,-689515067 ,-1002380572 ,-1025324604 ,
 -910042648 ,1087817280 ,-472259008 ,-908737717 ,1799411673 ,
 -974993963 ,-471703488 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
 0 ,0 ,0 ,0 ,0 ,0 ,1 ,
```

[1]See Problem Report number 335.

```
*
 ITMEVE=          0,
 IVDT=5 ,7 ,-1052712221 ,-690362304 ,-691650349 ,-689584789 ,
 1086440677 ,-975838236 ,-641373376 ,-1042955200 ,-473572903 ,
  909978683 ,1088864729 ,455144000 ,1549556800 ,-960118079 ,
 1086838483 ,-1002415012 ,1547714624 ,356532288 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,
.
```

Variables IUNIT(100) and IWEATHER control the routing of the alert messages for all alerts associated with a unit number (1-100) and weather alerts, respectively. The values of IUNIT and IWEATHER will cause an alert associated with a particular unit, or a weather alert, to go to the console using the standard scheme as follows:

Value	Console Alert is Sent to
1	1
2	2
3	3
4	1 & 3
5	2 & 3
6	1 & 2
7	1, 2, & 3

Variable MHOWSEND(30) controls the sending of alerts to Super Bees and the saving of alerts for the alerts post processor. Each of the 120 bytes (30 X 4) corresponds to an alert type (1st byte to alert type 1, 2nd byte to type 2, etc.), and the value of the byte determines what happens to that alert type every time the math model calls the alerts submodule to send a particular alert type. The values and results are as follows:

Value of MHOWSEND byte	Action on the Alert
0	Send to Super Bees and write on post-processor file.
1	Write on post-processor file, but send to Super Bee only after 1st minute of game.

2	Write on post-processor file. Don't send to Super Bee.
3	Don't send and don't write on post-processor file.

Thus, MHOWSEND allows selective control of each alert by type. Note that alerts generated by calling ALNUALRT are always sent to the Super Bees and always written to the post-processor file (these alerts are automatically assigned a type number of 1000). MHOWSEND can be namelisted in the namelist portion of the math model run deck (See Appendix A of the Data Base/Operations Manual) if the user desires to temporarily turn on or off an alert for testing purposes. (Normally, value is set in data statements in subroutine FORMAIN). Of course, care must be exercised in setting the value of a particular byte, since the smallest unit Xerox allows to be namelisted is 1 word (4 bytes).

Another variable, MTYPOFCM(35) controls alerts due to control measure crossings. Each of the 140 bytes of this array corresponds to a control measure type (1st byte to type 1 - assault line, 2nd byte to type 2 - fire coordination line, etc.) and the number in the byte controls the alerts. Also, the MTYPOFCM array is part of a scheme that stores the alphanumeric names of the control measures in a minimal amount of core. The legal values for this array will illustrate both:

VALUE IN MTYPOFCM	MEANING
0	Don't check for crossing of control measure by either fire or units. Thus, no alert will be generated for this type of control measure. Obviously, not checking saves running time as opposed to merely throwing an alert away after it has been generated.
1 to 40	Means an alert of type M will be generated whenever a unit that it applies to crosses it. Type M refers to the control measure types as they are in array NAMECM. For example, for control measure type #3 - line of contact - byte 3 of MTYPOFCM contains 10, and the 10th name in NAMECM is "CONTACT". How the "LINE" gets added is discussed below.

(1 to 40) + 64	Means generate alert type M as arranged in NAMECM (the 64 is subtracted to get M) whenever a unit to which the control measure applies fires support fire weapons across the control measure.
(1 to 40) + 128	Means generate alert type M as arranged in NAMECM (the 128 is subtracted to get M) whenever a unit to which the control measure applies fires support weapons short of the control measure.

Array MTYPOFCM accomplishes the above control indirectly by loading the right-most byte of the first word of each control measure when the control measure is created (in subroutine CONTMS). When a control measure is created interactively or from the data base, the information originally put in for the right-most byte (byte 3) of the first word of each control measure is converted as shown below:

INPUT VALUE	*MEANING*	*CONVERTED*	*TO*	*MEANING*
0	Do generate alerts	By subroutine CONTMS using MTYPOFCM	1-168	Do generate alerts, as shown above
non-zero (9 normally used on data base)	Don't generate alerts	By subroutine CONTMS	0	Don't generate alerts, this type of control measure for display only

Array MTYPOFCM can be namelisted in the run deck (See Appendix A of the Data Base/Operations Manual), just as MHOWSEND can, but both are normally set using the data statements in subroutine FORMAIN. Array ICMBREAK(6) and LABELCM(6) add the "LINE", "AREA", etc. to the alert. Each of the 6 positions in ICMBREAK determine which of the 6 four letter labels will be included with the name (NAMECM) of the control measures. The current values of ICMBREAK from the data statement in subroutine FORMAIN of 8, 18, 25, 29, 31, & 33 mean that the first 8 types listed (1-8) in NAMECM shall receive no additional label, 8 to 17 will receive the 1st label from LABELCM of "LINE", 18 to 24 will be labeled "AREA", 25-28 will be labeled "ZONE", 29 to 30 will be labeled "POSN", 31 to 32 will be labeled "BASE", and greater than 32 will be labeled "PNT". Thus, new types of control measures can be added to CATTS by merely changing a number of data statements. Note that there are data statements in the foreground code that must also be changed so that the new type would appear on the control measure menus.

A totally separate part of the alerts submodule is the alerts post-processor. This operates basically in the following manner: the alerts are saved during the game on a file (variable MHOWSEND, if desired, can cause some alert types to not be saved), and after a game has been terminated the operator can then run the alerts post-processor program PRALERTS (See Appendix B of the <u>Data Base/Operations Manual</u> for SOP #7, Terminate Game), and PRALERTS will sort and print the alerts for post-game analysis, critique, or as a permanent record of a game. The way that the alerts are sorted, and which alerts appear in which reports is very flexible, and is controlled by the card image file ANAMLIST. A report is a collection of alert types, sorted by time and unit. Following is a discussion of which variables control the reports, and how the alerts to be included in each report are controlled.

<u>Report/Case</u> - One report is generated for each data case (data cases are on file DB, ANAMLIST in card image). Data cases may be stacked back to back in order to generate multiple reports. (There are 4 data cases in the sample shown).

<u>Report Controls</u>

<u>Definition of alerts to be included in report</u>

The alerts to be included in each report are controlled by the following variables:

IACAT (120,4) - category array

IACAT(I, J) = 1 says to include alert number I in category J.

IACAT(I, J) = 0 excludes alert number I from category J.

IWANT (4) - defines category/categories to be included in this report.

IWANT(J) = 1 says to include category J in this report.

IWANT(J) = 0 excludes category J from this report.

Thus, for example, IWANT = 1, 0, 0, 1 says to include in this report all alerts in categories 1 and 4 as defined by the IACAT array.

IWANTS - flag to include or exclude special alerts (those alerts with alert number greater than 120) from this report.

IWANTS = 1, include in report

IWANTS = 0, exclude from report

Title to be printed on top of each page of report

ITITLE (20) - 80 characters. Nominalized to: ALERTS SORTED BY UNIT

Names to be used for alerts which do not have units associated with them

NAMEW (3) -	12 characters. Name associated with weather alert. In Data statement, nominalized to: WEATHER REP.
NAMEU (3) -	12 characters. Name associated with unattended ground sensors alert. In Data statement, nominalized to: GROUND SENS.
NAMES (3) -	12 characters. Name associated with special alerts (those alerts with alert number greater than 120). In Data statement, nominalized to: SPECIAL REP.

Number of lines per print page

NLPAGE - In Data statement, nominalized to 38.

Print order of alerts

For print purposes, there are four types of alerts:

1) alert having unit name associated with it
2) weather alert
3) unattended ground sensors alert
4) special alert (alert number greater than 120)

Each of these types has a sort priority assigned to it, which is used to make up the sort key. They are (with nominal values in parentheses).

IPRNAME (0)	names
IPRWEAT (500)	weather
IPRUNAT (501)	ground sensors
IPRSPEC (502)	special

The sort key priority for each named alert is computed as KEYX = IPRNAME + N where N is the table lookup index (i.e., Nth name). The nominal values cause the alerts to sort in the order: named, weather, ground sensors, and special. By changing the priority numbers, the user can change the sort order of the four types of alerts. Since there are up to 150 named units, the priority assigned to any alert type which is to follow the named alerts should be a value at least 151 greater than IPRNAME.

Processing associated with certain alert types

1) Weather alert. When alert type IANWEAT (nominalized in Data statement to 35) is encountered, priority IPRWEAT and name NAMEW (3) are associated with it. The user can cause this alert type to be treated like a named alert (name obtained from first item; up to 12 characters) by setting IANWEAT = 0.

2) Unattended ground sensors alert. When alert type IANUNAT (nominalized in Data statement to 64) is encountered, priority IPRUNAT and name NAMEU (3) are associated with it. The user can cause this alert type to be handled like a named alert by setting IANUNAT = 0.

3) Alert type IANTERM (nominalized in Data statement to 120) acts as an end-of-file. The user can force the program to keep reading until a zero alert type is found by setting IANTERM = 0.

Non-standard alerts

Non-standard alert numbers are defined by the array NSFLAG(120). This array is set up by subroutine NSALERT which reads FORTRAN unit F:12. The user may override entries in NSFLAG through the NAMELIST input.

```
NSFLAG(I) = 0, declares alert I to be standard
          = -1, declares alert I to be non-standard, and indicates that
                time item number must be looked up in alert text library.
          = N (positive), declares alert I to be non-standard, and
            indicates that Nth item of alert message of this type is
            time item.
```

Desired order of unit names

The order that unit names are to be sorted in is defined by the array NAMEUN (3, NNAMES) where NNAMES $\leq$ 150. This array is set up by subroutine ORDERUN which reads FORTRAN unit 11. The user may override this information by entering a list of names in the desired order through TEXTINP input (see further on). The program will run, with the names sorted on a first in/first out basis, with no list of names, that is, with NNAMES = 0. This means the order the alerts appear on the reports will be constructed from the order that the reports are on the post-processor file, and once this order is constructed it will be used for the remainder of the reports.

Last case indicator

```
ITERM = 0, there is another case
      = 1, stop after finishing this case
```

Files referenced (F: # - see RBM Reference Manual on ASSIGN statements)

```
105 - card reader
108 - printer
 10 - input alert messages
 11 - desired order unit names
 12 - non-standard alert types
 17 - alert text library
```

Case Data

Subroutine NSALERT which sets up the non-standard alert numbers, and subroutine ORDERUN which sets up the desired order of unit names are called only before the first case. Also, all nominal values, as discussed in section II, are not reset before multiple cases. Thus, any values that are changed by the user through input, remain changed through multiple cases.

Each case of input consists of two types of data:

1) Hollerith input processed by subroutine TEXTDVP and referenced as TEXTINP.

2) Numeric data, read in by NAMELIST.

TEXTINP data

Card is broken into three fields:

a) Col. 1-10, variable name and optional subscript. Variable name must precede subscript and there must be at least one blank between the name and the subscript. The name does not have to be left justified. For example: IACAT 2.

   A continuation of data from the previous line is indicated by leaving col. 1-10 blank. The absence of a subscript defaults to a subscript of 1.

b) Col. 11-70, data field. Format depends on variable being entered.

c) Col. 71-80, comments. Ignored.

Data enterable through TEXTINP

All variable entries are optional and may be entered in any order except for the end-of-data terminator END which must be entered and which must be last.

a) Category array, IACAT
   This array may be set up through NAMELIST, but it is easier to do it through TEXTINP.

   Input format

1	11 70	
IACAT i	include/exclude designations for alerts 1-60	card
	include/exclude designations for alerts 60-120	next card

   i = alert report category number 1, 2, 3, or 4

   A non-blank character in a column, indicates that the corresponding alert number is to be included in category (i); a blank excludes the alert.

   The data must be entered separately for each category. Thus, the data for each category must be explicitly preceded by IACAT i.

   The program is set up to accept IANMAX entries for each category, so in the future if IANMAX is changed to a number larger than 120, the processor will still function properly.

b) Report title array ITITLE
   Input format

1	11	30	70
ITITLE	first 60 characters		
	last 20 characters		

   The continuation line may be omitted, but then the last 20 characters will remain whatever they were before.

The first 20 characters of the nominal ITITLE are blank, so if the title is not long, the user can enter:

1	11 … 70
ITITLE 6	last 60 characters, first 20 left as they were

c) Names to be printed on report for:
  weather alert - NAMEW
  unattended ground sensor alert - NAMEU
  special alert (alert number greater than 120) - NAMES

Input format

1	11	22
NAMEW	weather alert name	

NAMEU	ground sensor alert name	

NAMES	special alert name	

d) Unit names desired order array, NAMEUN

Input format

1	11	22	31	42	51	62
NAMEUN i	12 character unit name	▨	12 character unit name	▨	12 character unit name	

Any 12 character data field that is all blank is ignored. If these entries extend the length of NAMEUN, the length NNAMES is automatically updated.

Example:

nameun 121		▨	AAAAAAAAAAAA	▨	BBBBBBBBBBBB	

	CCCCCCCCCCCC	▨		▨	DDDDDDDDDDDD	

Result NAMEUN (1, 121) = AAAAAAAAAAAA
  NAMEUN (1, 122) = BBBBBBBBBBBB
  NAMEUN (1, 123) = CCCCCCCCCCCC
  NAMEUN (1, 124) = DDDDDDDDDDDD

e) End of Hollerith data terminator. Must be entered and must be last. (That is, must immediately precede the NAMELIST data.)

Input format

1	11
END	

Data enterable through NAMELIST

The main program contains a NAMELIST statement with no parameters so most of the variables in the program can have values assigned through NAMELIST input. The only parameters normally entered are:

a) Category/categories to be included in this report array, IWANT
IWANT (I) = 1, include category I
= 0, exclude category I

b) Include/exclude(1/0) special alerts (alert numbers greater than 120) in this report flag, IWANTS.

c) Last case flag, ITERM.
Set ITERM = 1 in last case, to cause program to terminate

d) Number of lines per report page, NLPAGE. Set in first case to adjust for larger paper.

e) Sort priority numbers which affect order that different types of alerts sort in.

IPRNAME - named
IPRWEAT - weather
IPRUNAT - unattended ground sensors
IPRSPEC - special

SAMPLE COPY OF FILE DB, ANAMLIST

```
IACAT 1   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
END
    IWANT = 1, 0, 0, 0,
    IWANTS = 1,
*
ITITLE 6                ENGAGEMENT ALERTS SORTED BY TIME
IACAT 2   XX  X               XX  X          X                XX  XXXX ENGAGE
          XXXX XXXXXXX   X X  XXXX                                  X  ENGAGE 2
END
 IWANT = 0, 1, 0, 0,
  IWANTS = 0,
*
ITITLE 6             LOSS REPORT ALERTS SORTED BY TIME
IACAT 3       X                                                  XXX   LOSS 1
                X       X  XX                                       X  LOSS 2
END
  IWANT = 0, 0, 1, 0,
*
ITITLE 6             RESUPPLY ACTIONS ALERTS SORTED BY UNIT
IACAT 4                                                                RESUP. 1
                                        XX                          X  RESUP. 2
END
  IWANT = 0, 0, 0, 1,
   ITERM = 1,
*
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST on Previous Page

```
              ALERTS SORTED BY UNIT                                                      PAGE      1
  UNIT        TIME                MESSAGE

2/A/1/107     080519  2/A/1/107     VU 83992794 080519 FIRING AT AIR UNIT ABLE09

              080520  2/A/1/107     VU 83962792 080520 CEASING FIRE AT AIR UNIT ABLE09

              080535  2/A/1/107     VU 83462770 080535 RECEIVING FIRE FROM  1 PLT
                                    OF TYPE SMALL-ARMS, AT A RANGE OF 963 M
                                    FIRING: NONE

              080540  2/A/1/107 VISUALLY DETECTED WPNS/B/2-77 VU 83102748 080540

                      2/A/1/107     VU 83312763 080540 RECEIVING FIRE FROM  2 PLT
                                    OF TYPE SMALL-ARMS  MORTARS, AT A RANGE OF 248 M
                                    FIRING: SMALL-ARMS

                      2/A/1/107     VU 83312763 080540 FIRING STARTED 080535 AND ENDED (NOT ENDED)
                                     PERSONNEL    1

3/A/1/107     080523  3/A/1/107 VISUALLY DETECTED 1/B/2-77 VU 80522893 080523

              080524  3/A/1/107 VISUALLY DETECTED HV MORT/2-77 VU 86592771 080524

              080530  3/A/1/107 VISUALLY DETECTED 3/B/2-5 VU 80412893 080530

1/B/1/107     080515  1/B/1/107 VISUALLY DETECTED 3/A/2-5 VU 86592168 080515

              080520  1/B/1/107 VISUALLY DETECTED 3/B/2-5 VU 80642889 080520

2/B/1/107     080525  2/B/1/107 VISUALLY DETECTED 3/B/2-5 VU 80412893 080525
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST

```
                      ALERTS SORTED BY UNIT                                                   PAGE      2
 UNIT          TIME                 MESSAGE

               080529  2/B/1/107 VISUALLY DETECTED 3/A/2-5 VU 86592168 080529

3/B/1/107      080519  3/B/1/107      VU 81972239 080519 FIRING AT AIR UNIT ABLE09

                       3/B/1/107      VU 81972239 080519 CEASING FIRE AT AIR UNIT ABLE09

2/C/1/107      080519  2/C/1/107      VU 79592766 080519 CAS FM AIR STRIKE
                                      PERSONNEL      14 BMP/SAGGER      2

                       2/C/1/107      VU 79592766 080519 FIRING STARTED 080519 AND ENDED (NOT ENDED)
                                       PERSONNEL    14 BMP/SAGGER     2

               080523  2/C/1/107      VU 79592766 080523 FIRING AT AIR UNIT ABLE09

               080524  2/C/1/107      VU 79592766 080524 CEASING FIRE AT AIR UNIT ABLE09

               080536  2/C/1/107      VU 79592766 080536 RECEIVING FIRE FROM  1 CO
                                      OF TYPE ARTILLERY, AT A RANGE OF 9235 M
                                      FIRING: NONE

               080539  2/C/1/107      VU 79592766 080539 FIRING AT AIR UNIT ABLE07

                       2/C/1/107      VU 79592766 080539 CEASING FIRE AT AIR UNIT ABLE07

                       2/C/1/107      VU 79592766 080539 FIRING STARTED 080519 AND ENDED 080539
                                       PERSONNEL    14 BMP/SAGGER     2

               080540  2/C/1/107      VU 79592766 080540 FIRING AT AIR UNIT ABLE07

                       2/C/1/107      VU 79592766 080540 CAS FM AIR STRIKE
                                      PERSONNEL       2
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                 ALERTS SORTED BY UNIT                     .                              DIRECTORY   PAGE      9
         UNIT                PAGE           UNIT            PAGE             UNIT          PAGE
  1  2/A/1/107                 1      14  2/A/2-77            4        26  1/B/2-4           6
  2  3/A/1/107                 1      15  1/B/2-77            4        27  3/B/2-4           6
  3  1/B/1/107                 1      16  2/B/2-77            4        28  1/C/2-4           6
  4  2/B/1/107                 1      17  3/B/2-77            5        29  SCT PLT/2-4       6
  5  3/B/1/107                 2      18  WPNS/B/2-77         5        30  1/B/2-5           7
  6  2/C/1/107                 2      19  CD GP/C/2-77        5        31  2/B/2-5           7
  7  3/A/2/107                 3      20  2/C/2-77            5        32  3/B/2-5           7
  8  3/C/2/107                 3      21  WPNS/C/2-77         5        33  SCT PLT/2-5       7
  9  MORT/2/107                3      22  HV MORT/2-77        6        34  HVY MORT/2-5      7
 10  RECON/107                 3      23  1/A/2-4             6        35  A/7-50            8
 11  HOW BTRY/107              4      24  2/A/2-4             6        36  ABLE09            8
 12  BN CP                     4      25  3/A/2-4             6        37  ABLE07            8
 13  1/A/2-77                  4
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                                  ENGAGEMENT ALERTS SORTED BY TIME                                  PAGE     1
   UNIT         TIME                    MESSAGE

2/A/1/107      080519  2/A/1/107    VU 83992794 080519 FIRING AT AIR UNIT ABLE09

               080520  2/A/1/107    VU 83962792 080520 CEASING FIRE AT AIR UNIT ABLE09

               080535  2/A/1/107    VU 83462770 080535 RECEIVING FIRE FROM  1 PLT
                                    OF TYPE SMALL-ARMS, AT A RANGE OF 963 M
                                    FIRING: NONE

               080540  2/A/1/107 VISUALLY DETECTED WPNS/B/2-77 VU 83102748 080540

                       2/A/1/107    VU 83312763 080540 RECEIVING FIRE FROM  2 PLT
                                    OF TYPE SMALL-ARMS  MORTARS, AT A RANGE OF 248 M
                                    FIRING: SMALL-ARMS

                       2/A/1/107    VU 83312763 080540 FIRING STARTED 080535 AND ENDED (NOT ENDED)
                                     PERSONNEL     1

3/A/1/107      080523  3/A/1/107 VISUALLY DETECTED 1/B/2-77 VU 80522893 080523

               080524  3/A/1/107 VISUALLY DETECTED HV MORT/2-77 VU 86592771 080524

               080530  3/A/1/107 VISUALLY DETECTED 3/B/2-5 VU 80412893 080530

1/B/1/107      080515  1/B/1/107 VISUALLY DETECTED 3/A/2-5 VU 86592168 080515

               080520  1/B/1/107 VISUALLY DETECTED 3/B/2-5 VU 80642889 080520

2/B/1/107      080525  2/B/1/107 VISUALLY DETECTED 3/B/2-5 VU 80412893 080525
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                              ENGAGEMENT ALERTS SORTED BY TIME                          PAGE    2
 UNIT          TIME                 MESSAGE

               080529  2/B/1/107 VISUALLY DETECTED 3/A/2-5 VU 86592168 080529

3/B/1/107      080519  3/B/1/107    VU 81972239 080519 FIRING AT AIR UNIT ABLE09

                       3/B/1/107    VU 81972239 080519 CEASING FIRE AT AIR UNIT ABLE09

2/C/1/107      080519  2/C/1/107    VU 79592766 080519 CAS FM AIR STRIKE
                                    PERSONNEL      14 BMP/SAGGER      2

                       2/C/1/107    VU 79592766 080519 FIRING STARTED 080519 AND ENDED (NOT ENDED)
                                     PERSONNEL    14 BMP/SAGGER     2

               080523  2/C/1/107    VU 79592765 080523 FIRING AT AIR UNIT ABLE09

               080524  2/C/1/107    VU 79592766 080524 CEASING FIRE AT AIR UNIT ABLE09

               080536  2/C/1/107    VU 79592766 080536 RECEIVING FIRE FROM  1 CO
                                    OF TYPE ARTILLERY, AT A RANGE OF 9235 M
                                    FIRING: NONE

               080539  2/C/1/107    VU 79592766 080539 FIRING AT AIR UNIT ABLE07

                       2/C/1/107    VU 79592766 080539 CEASING FIRE AT AIR UNIT ABLE07

                       2/C/1/107    VU 79592766 080539 FIRING STARTED 080519 AND ENDED 080539
                                     PERSONNEL    14 BMP/SAGGER     2

               080540  2/C/1/107    VU 79592766 080540 FIRING AT AIR UNIT ABLE07

                       2/C/1/107    VU 79592766 080540 CAS FM AIR STRIKE
                                    PERSONNEL       2
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                             ENGAGEMENT ALERTS SORTED BY TIME                  DIRECTORY   PAGE     9
      UNIT              PAGE              UNIT          PAGE              UNIT          PAGE
  1   2/A/1/107           1         14   2/A/2-77         4         26   1/B/2-4          6
  2   3/A/1/107           1         15   1/B/2-77         4         27   3/B/2-4          6
  3   1/B/1/107           1         16   2/B/2-77         4         28   1/C/2-4          6
  4   2/B/1/107           1         17   3/B/2-77         5         29   SCT PLT/2-4      6
  5   3/B/1/107           2         18   WPNS/B/2-77      5         30   1/B/2-5          7
  6   2/C/1/107           2         19   CD GP/C/2-77     5         31   2/B/2-5          7
  7   3/A/2/107           3         20   2/C/2-77         5         32   3/B/2-5          7
  8   3/C/2/107           3         21   WPNS/C/2-77      5         33   SCT PLT/2-5      7
  9   MORT/2/107          3         22   HV MORT/2-77     6         34   HVY MORT/2-5     7
 10   RECON/107           3         23   1/A/2-4          6         35   A/7-50           8
 11   HOW BTRY/107        4         24   2/A/2-4          6         36   ABLE09           8
 12   BN CP               4         25   3/A/2-4          6         37   ABLE07           8
 13   1/A/2-77            4
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                              LOSS REPORT ALERTS SORTED BY TIME                        PAGE    1
  UNIT          TIME                      MESSAGE

2/A/1/107      080540  2/A/1/107     VU 83312763 080540 FIRING STARTED 080535 AND ENDED (NOT ENDED)
                                      PERSONNEL     1

2/C/1/107      080519  2/C/1/107     VU 79592766 080519 CAS FM AIR STRIKE
                                     PERSONNEL      14 BMP/SAGGER      2

                       2/C/1/107     VU 79592766 080519 FIRING STARTED 080519 AND ENDED (NOT ENDED)
                                      PERSONNEL   14 BMP/SAGGER     2

               080539  2/C/1/107     VU 79592766 080539 FIRING STARTED 080519 AND ENDED 080539
                                      PERSONNEL   14 BMP/SAGGER     2

               080540  2/C/1/107     VU 79592766 080540 CAS FM AIR STRIKE
                                     PERSONNEL      2

                       2/C/1/107     VU 79592766 080540 FIRING STARTED 080540 AND ENDED (NOT ENDED)
                                      PERSONNEL     2

3/A/2/107      080519  3/A/2/107     VU 83951842 080519 FIRING STARTED 080412 AND ENDED (NOT ENDED)
                                      PERSONNEL     0 SA-7             1

               080535  3/A/2/107     VU 83951842 080535 FIRING STARTED 080412 AND ENDED (NOT ENDED)
                                      PERSONNEL     1

MORT/2/107     080515  MORT/2/107    VU 81601811 080515 FIRING STARTED 080350 AND ENDED (NOT ENDED)
                                      PERSONNEL     5 TRUCK            1
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                                 LOSS REPORT ALERTS SORTED BY TIME                                PAGE     2
  UNIT           TIME                    MESSAGE

HQA BTRY/107   080516  HQA BTRY/107  VU 72192308 080516 FIRING STARTED 080437 AND ENDED (NOT ENDED)
                                     PERSONNEL     5 TRUCK          1

2/C/2-77       080521  2/C/2-77      VU 83621848 080521 AMMUNITION LOW

1/B/2-4        080516  1/B/2-4       VU 83222049 080516 FIRING STARTED 080428 AND ENDED (NOT ENDED)
                                     PERSONNEL     2 M-60/105 MM    1

3/B/2-4        080524  3/B/2-4       VU 83622044 080524 FIRING STARTED 080350 AND ENDED (NOT ENDED)
                                     PERSONNEL     3 M-60/105 MM    1

1/C/2-4        080525  1/C/2-4       VU 83222049 080525 FIRING STARTED 080350 AND ENDED (NOT ENDED)
                                     PERSONNEL     2 M-60/105 MM    1

SCT PLT/2-5    080516  SCT PLT/2-5   VU 72472194 080516 FIRING STARTED 080402 AND ENDED (NOT ENDED)
                                     PERSONNEL     3 M-113/.50CAL   1
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                          LOSS REPORT ALERTS SORTED BY TIME                    DIRECTORY    PAGE      3
      UNIT           PAGE              UNIT         PAGE               UNIT        PAGE
  1  2/A/1/107         1           5  HOW BTRY/107    2             8  3/B/2-4        2
  2  2/C/1/107         1           6  2/C/2-77        2             9  1/C/2-4        2
  3  3/A/2/107         1           7  1/B/2-4         2            10  SCT PLT/2-5    2
  4  MORT/2/107        1
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                              RESUPPLY ACTIONS ALERTS SORTED BY UNIT                        PAGE     1
                 TIME                        MESSAGE

1/A/1/107       060330  1/A/1/107    VU 29803030 060330 40 AKM   TO 107TH REGT

                        1/A/1/107    VU 29803030 060330 1 SAGGER/BMP   TO 107TH REGT

                        1/A/1/107    VU 29803030 060330 1 73MM/BMP   FM 107TH REGT

                        1/A/1/107    VU 29803030 060330 40 73MM SHELL AMMO FM 107TH REGT

                        1/A/1/107    VU 29803030 060330 40 AKM   TO 107TH REGT

                        1/A/1/107    VU 29803030 060330 1 SAGGER/BMP   TO 107TH REGT

                        1/A/1/107    VU 29803030 060330 1 73MM/BMP   FM 107TH REGT

                        1/A/1/107    VU 29803030 060330 40 73MM SHELL AMMO FM 107TH REGT

                060609  1/A/1/107    VU 59182532 060609 1 73MM/BMP   TO 107TH REGT

                        1/A/1/107    VU 59182532 060609 1 SAGGER/BMP   TO 107TH REGT


2/A/1/107       060330  2/A/1/107    VU 29803090 060330 27 AKM   TO 107TH REGT

                        2/A/1/107    VU 29803090 060330 1 SAGGER/BMP   TO 107TH REGT

                        2/A/1/107    VU 29803090 060330 1 73MM/BMP   FM 107TH REGT

                        2/A/1/107    VU 29803090 060330 40 73MM SHELL AMMO FM 107TH REGT

                        2/A/1/107    VU 29803090 060330 27 AKM   TO 107TH REGT

                        2/A/1/107    VU 29803090 060330 1 SAGGER/BMP   TO 107TH REGT

                        2/A/1/107    VU 29803090 060330 1 73MM/BMP   FM 107TH REGT
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                         RESUPPLY ACTIONS ALERTS SORTED BY UNIT                         PAGE    2
             TIME                       MESSAGE

                      2/A/1/107   VU 29803090 060330 40 73MM SHELL AMMO FM 107TH REGT

3/A/1/107    060330   3/A/1/107   VU 29203050 060330 3 TRUCK  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 40 AKM  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 1 SAGGER/BMP  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 1 73MM/BMP  FM 107TH REGT
                      3/A/1/107   VU 29203050 060330 40 73MM SHELL AMMO FM 107TH REGT
                      3/A/1/107   VU 29203050 060330 3 TRUCK  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 40 AKM  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 1 SAGGER/BMP  TO 107TH REGT
                      3/A/1/107   VU 29203050 060330 1 73MM/BMP  FM 107TH REGT
                      3/A/1/107   VU 29203050 060330 40 73MM SHELL AMMO FM 107TH REGT

1/B/1/107    060330   1/B/1/107   VU 29002960 060330 39 AKM  TO 107TH REGT
                      1/B/1/107   VU 29002960 060330 1 SAGGER/BMP  TO 107TH REGT
                      1/B/1/107   VU 29002960 060330 1 73MM/BMP  FM 107TH REGT
                      1/B/1/107   VU 29002960 060330 40 73MM SHELL AMMO FM 107TH REGT
                      1/B/1/107   VU 29002960 060330 39 AKM  TO 107TH REGT
```

Sample ALERT Printout Produced by Copy of File DB ANAMLIST (Cont'd)

```
                         RESUPPLY ACTIONS ALERTS SORTED BY UNIT                    DIRECTORY   PAGE    27
        UNIT                   UNIT                      UNIT                      UNIT
  1  1/A/1/107          14  2/B/2/107          27  A/1/79              40  1/C/2-77
  2  2/A/1/107          15  3/B/2/107          28  B/1/79              41  CSC
  3  3/A/1/107          16  1/C/2/107          29  C/1/79              42  SCT PLT/2-77
  4  1/B/1/107          17  2/C/2/107          30  UNIT 47             43  AT/2-77
  5  2/B/1/107          18  3/C/2/107          31  UNIT 54             44  1/C/1-23
  6  3/B/1/107          19  A/3/107            32  UNIT 80             45  2/C/1-23
  7  1/C/1/107          20  B/3/107            33  1/A/2-77            46  3/C/1-23
  8  2/C/1/107          21  C/3/107            34  2/A/2-77            47  A/7-50
  9  3/C/1/107          22  C/TK/107           35  3/A/2-77            48  B/7-50
 10  1/A/2/107          23  HOW BTRY/107       36  WPNS/A/2-77         49  C/7-50
 11  2/A/2/107          24  HOW BN/CAA         37  1/B/2-77            50  155/7-52
 12  3/A/2/107          25  HOW BN/17F         38  2/B/2-77            51  203/1-43
 13  1/B/2/107          26  GH BN/CAA          39  3/B/2-77
STOP* 0
```

5.7.2.2 Assumptions and Data Sources

The obvious assumption inherent in this submodule is that useful information can be conveyed to the controllers during a game using the alerts. This has proven to be true with one constraint: volume. If there are too many alerts of a given type, there might as well not be any alerts of the type. A good example is the visual detection alerts. Sheer volume causes most of these alerts to be ignored by the controllers, and when a controller does wish to know if one unit has detected another, this information is lost in the hundreds of detection alerts.

All data for the variables which control the alerts was provided by the Army CATTS project personnel at Fort Benning.

5.7.2.3 Equations

There are no equations inherent to this submodule. The equations which govern the alerts are covered in their respective write-ups.

5.7.3 RAM Alerts Submodule

5.7.3.1 Operation

RAM alerts are generated in the various modules by a call to subroutine RAMALERT. This causes a message to be sent via the RAMTEK to the lower 1/3 of the selected instructor's TV screen. The instructor must read the alert and select the IGNORE in the lower right hand corner of the alert before he can use the menus or move the TV camera to look at a different portion of the map. RAM alerts are error messages or instructions resulting from simulation control or use of one of the command and control menus. A complete, though slightly out of date, list of all RAM alerts is contained in Section 5.7.3.3 of the Programming Report and Section 3.109 of this User's Manual. These alerts are displayed immediately after they are generated on the TV monitors. The instructor #, the number of characters in the message, and the name of the array that contains the message or the array name including indices of the first element of the array, or a Hollerith character string, are the arguments in a call to RAMALERT, which appears as shown below:

```
CALL RAMALERT(INS#, NUMBCHAR, MESGARAY)
                    where

                          INS# is instructor number
                              (1,2, or 3)
                          NUMBCHAR is the total number of
                          characters in the message.
                          MESGARAY is the name of the array
                          which contains the message.
```

Note that a colon (:) appearing in the message will cause the characters after it to appear on the next line down the TV screen, but the colon itself will not appear on the screen. Thus, no RAM alert message should contain a colon unless a next line is desired. Also, if the INS# is an illegal value (other than 1, 2, or 3) the message will appear at console 1 automatically. Table 5-51 gives a brief description of each routine in this submodule. However, since this submodule does not contain any subroutines (RAMALERT is part of the foreground software), Table 5-51 is presented for consistency and completeness rather than information content.

Table 5-51. RAM Alerts Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
RAMALERT	(This is a foreground software subroutine--see Section 4.2.1.2.1 in the Trainer Programming Report (pages 4-75 thru 4-77) for a description of this subroutine.)	

5.7.3.2 Assumptions and Data Sources

The underlying assumption of the RAM alerts is that the controllers should be informed immediately, in a manner they can not miss seeing, of errors made: when the command and control events are being processed, when the model is being initialized (SAVE files not found, for example), and of pertinent instructions or errors when simulation control is being performed. The only potential problem with RAM alerts would be the generation of too many RAM alerts. This would then cause the controllers to be so busy processing RAM alerts that they would be unable to use the console to input command and control or move the camera. Thus, during the development of CATTS, every effort was made to keep the number of RAM alerts to a minimum, while still conveying the necessary information to the controllers.

The source of all the RAM alerts is TRW engineering and design.

5.7.3.3 Equations

All the equations which generate RAM alerts are merely checking to make sure indices to arrays have not been exceeded, legal numbers have been received from the foreground for a particular portion of an event, or that some model variable has been set. What caused the RAM alert should be obvious from the explanation in Section 3.109 of this User's Manual.

5.7.4 RATT Submodule

5.7.4.1 Operation

This submodule allows the controllers to send messages from the Super Bee terminals to the simulated RATT teletype (ASR-35) in the TOC. A controller may edit and send the current alert message on the Super Bee screen, may edit and send any of the canned RATT messages (up to 100), or may compose a new message and send it. The controller may also "can", or save on disk, any message he desires. The procedures for all of these are described in detail in Section 2.7 of the CATTS Operator's Manual.

The players may type any message they wish on the ASR-35 and send it to the controllers. This procedure is also described in the CATTS Operator's Manual. Which controllers receive the messages from the TOC is determined by variable IRATT, which can be set in either the namelist portion of the data base scenario file (example-NBIG) or in the namelist portion of the run deck. This variable will route the messages as shown:

Value	Routing
0	Ignore - route to none of the consoles
1	Console 1
2	Console 2
3	Console 3
4	Console 1 & 3
5	Console 2 & 3
6	Console 1 & 2
7	Console 1, 2, & 3

Also of interest in the operation of the RATT is program RATTEDIT. This program allows the canned RATT messages (i.e., saved on disk) to be listed on the line printer (cataloged) and deleted. This program is explained in Appendix B of the Data Base/Operations Manual and Section 4.4.15.3.2 of the Programming Report.

5.7.4.2 Assumptions and Data Sources

The RATT submodule was designed to be as flexible as possible since the content of the RATT messages and the use of the RATT system are bound to change with new scenarios and changes in Standard Operating Procedures.

5.7.4.3 Equations

No equations are applicable to this submodule. All the code that runs this submodule is in the foreground.

5.7.5 Status Report Submodule

5.7.5.1 Operation

The Status Report Submodule prints the Army version of the math model status report (a TRW version is also available - see Section 5.7.6, Diagnostic and Miscellaneous Output) on the line printer. This submodule also outputs on the line printer the final game status report, which is part of the post game summary (Section 5.7.6 covers the other part).

Figure 5-102 shows the subroutine linkages for the Status Report Submodule and Table 5-52 gives a brief description of each routine as well as its principal inputs and outputs. As Figure 5-102 shows, the principal routine of the submodule is STATREP1. This routine is not called (therefore no status report is produced) if variable IDOSTAT is zero. If IDOSTAT is equal to a positive number N, then STATREP1 is called every N model minutes and again at the end of the exercise. See SOP #7 in Appendix B of Data Base/Operations Manual.

Once STATREP1 is called, each of the five major reports it produces is separately controlled via the input table IOINTRVL. Table 5-53 shows the effect of possible settings upon the output. Note that it is possible to print the different reports at different intervals, and even to intermix TRW and Army versions.

The status report format is the same for both the periodic and final reports. The only difference is that the change in levels shown are since the time of the last report on the periodic report, but on the final the change is the difference from the start of the game to the end of the game. There are five basic reports or parts of the status report and in the order printed they are: Unit, Equipment, Ammunition, Fuel, and Personnel. A sample of the first page of each of these reports is shown on the next page.

All the items on the reports are self-explanatory with the possible exception of the letters which appear on the Unit report under the heading CCFLAGS. The letters have the following meaning:

FLAG	MEANING
M	Unit is under maneuver command and control input via the menus and is moving mounted.

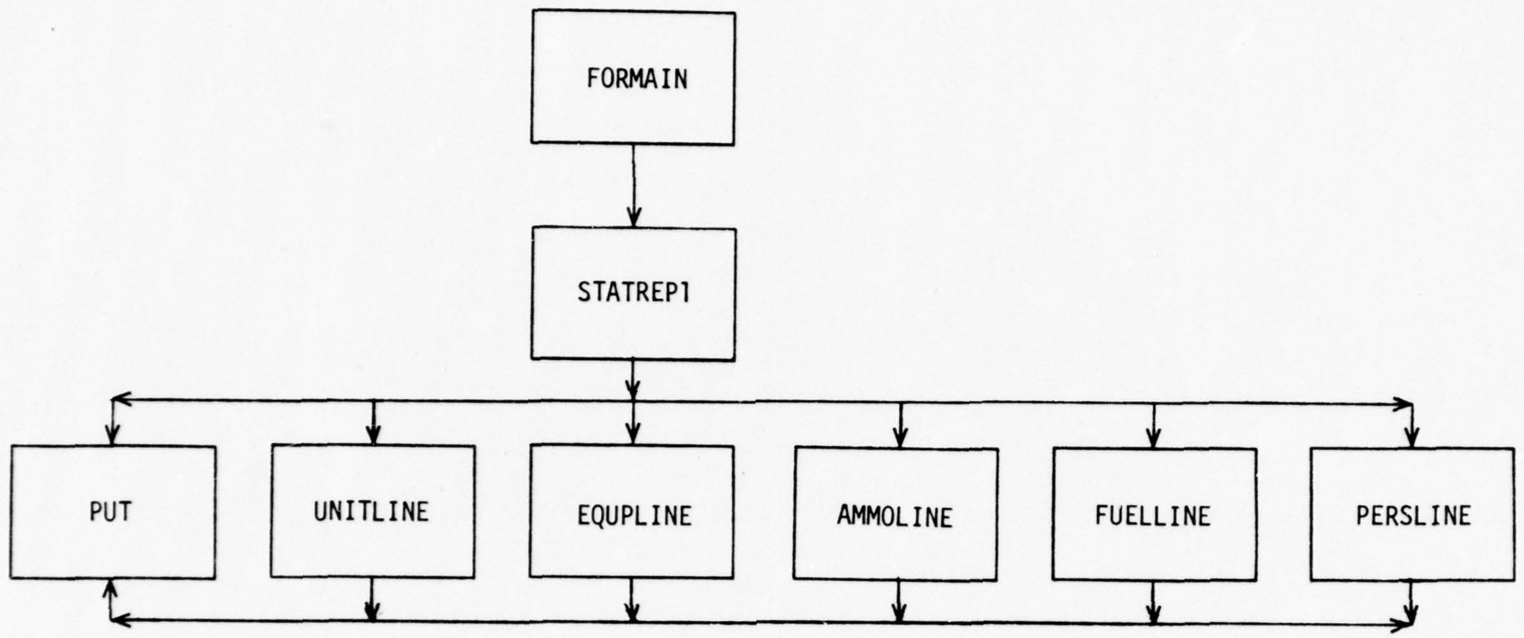


Figure 5-102. Subroutine Linkage of Status Report Submodule

Table 5-52. Status Report Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
STATREP1	Main driver for submodule. Cycles through all units for each report type and calls appropriate line formatting routine for those units that are active.	Input: None Output: (Prints Army status report on line printer.)
UNITLINE	Outputs a line of the unit status report section and saves the necessary information for the next report.	Input: None Output: (Prints, on line printer unit, portion of Army status report.)
EQUPLINE	Outputs the current equipment levels for a unit, computing the change since the last report and saving current levels for the next report.	Input: None Output: (Prints the equipment portion of the Army status report.)
AMMOLINE	Outputs the current ammunition levels for a unit, computing the change since the last report and saving the current levels for the next report.	Input: NETAMU(IU,J) - Current amount (in tenths of a round) of J-th ammo type carried by unit IU. Actual ammo type given by NTAMU(IU,J). ($1 \leq IU \leq 100$, $1 \leq J \leq 14$) NETAMX(JU,J) - Initial amount (in tenths of a round) of J-th ammo type carried by unit JU. Actual ammo type given by NTAMU(JU,J). ($1 \leq JU \leq 100$, $1 \leq J \leq 14$) NTAMU(IU,J) - Ammunition type number of J-th ammo type carried by unit IU. ($1 \leq IU \leq 100$, $1 \leq J \leq 14$) Output: (Prints ammunition portion of Army status report.)
FUELLINE	Outputs the current and change levels of fuel for one unit and saves the current levels.	Input: None Output: (Prints fuel portion of Army status report.)

Table 5-52. Status Report Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
PERSLINE	Outputs the current personnel levels for a unit for each type of personnel. The current levels are saved and the change since the last report is printed.	Input:	None	
		Output:	(Prints personnel portion of Army status report.)	
PUT	This subroutine is called by the printing routines to enter a line of print in the output buffer. It performs the necessary page computations, outputting page spacing and titles as necessary.	Input:	L	- Passes first word (4 characters) of line to be printed.
			N	- Number of characters on the line.
		Output:	KPAGE	- Status report page count.
			LPAGE	- Section page count.
			NCOUNT	- Number of lines on the current page.
PUT1	Called by subroutine PUT, this assembly language routine performs the character packing and pointer maintenance necessary to enter a print line into the output buffer.	Input:	L	- First argument--array containing line to be printed.
			N	- Second argument--number of characters on the line.
		Output:	None	
XPUT	This is an entry point in subroutine PUT1 that is called to transmit the output buffer to the foreground printing routine.	Input:	(See subroutine PUT1.)	
		Output:	(See subroutine PUT1.)	
COMMDEF	This is a dummy subroutine used to define the overlay segments used to contain the data from previous reports used to compute changes.	Input:	None	
		Output:	None	
B	This is a dummy assembly language routine used to define an external address for the output buffer. The address, B, is used by subroutine PUT1 as the beginning of in-core buffer address.	Input:	None	
		Output:	None	

Table 5-53. Controlling the Printing of Status Reports Individually

REPORT	CONTROLLING VARIABLE	RESULT	
UNIT STATUS	IOINTRVL(1)	$= 0$	Report never printed.
EQUIPMENT	IOINTRVL(2)	$= N > 0$	TRW version printed every N minutes. (See 5.7.6)
AMMUNITION	IOINTRVL(3)		
PERSONNEL	IOINTRVL(4)	$= N < 0$	Army version printed every -N minutes, but only if IDOSTAT set such that STATREP1 called every -N minutes.
FUEL	IOINTRVL(13)		

Sample Army Status Report

```
         UNIT STATUS REPORT
           TIME OF THIS REPORT (NOW)    050302
           TIME OF LAST REPORT (LAST)
--------------------------------------------------------------------------------------------------------------
                                                                           *-M=UNDER MANEUVER COMMAND & CONTROL
                                                                            -F=UNDER FIRE COMMAND & CONTROL
                                                                            -D=MOVING DISMOUNTED
--------------------------------------------------------------------------------------------------------------
            UNIT                     LOCATION               ELEVATION   DIRECTN   ON ROAD   PCT SUP   REDCON   CC-*
  NO.          NAME           NOW                LAST        NOW LAST   NOW LST   NOW LST   NOW LST   NOW LST  FLG
--------------------------------------------------------------------------------------------------------------
                            *                             *           *         *         *         *         *
                            *                             *           *         *         *         *         *
                            *                             *           *         *         *         *         *
--------------------------------------------------------------------------------------------------------------
OPGP   1 1/107 (RED)        * VU3010029957                *           *  90     *         *         *         *
       1 1/A/1/107          * VU2980030300                *  220      *  90     *         *  0      *  1      *
       2 2/A/1/107          * VU2980030900                *  213      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
       3 3/A/1/107          * VU2920030500                *  234      *  90     *         *  0      *  1      *
       4 1/B/1/107          * VU2900029600                *  210      *  90     *         *  0      *  1      *
       5 2/B/1/107          * VU2900030200                *  210      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
       6 3/B/1/107          * VU2850030000                *  212      *  90     *         *  0      *  1      *
       7 1/C/1/107          * VU2900030900                *  228      *  90     *         *  0      *  1      *
       8 2/C/1/107          * VU2900031600                *  225      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
       9 3/C/1/107          * VU2850031200                *  227      *  90     *         *  0      *  1      *
      10 MORT/1/107         * VU2980031700                *  225      *  90     *         *  0      *  1      *
                            *                             *           *         *         *         *         *
--------------------------------------------------------------------------------------------------------------
OPGP   2 2/107 (RED)        * VU2970029471                *           *  90     *         *         *         *
      11 1/A/2/107          * VU3010029000                *  225      *  90     *         *  0      *  1      *
      12 2/A/2/107          * VU3010028500                *  232      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
      13 3/A/2/107          * VU2960028800                *  230      *  90     *         *  0      *  1      *
      14 1/B/2/107          * VU2910029000                *  223      *  90     *         *  0      *  1      *
      15 2/B/2/107          * VU2910028400                *  241      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
      16 3/B/2/107          * VU2850028800                *  232      *  90     *         *  0      *  1      *
      17 1/C/2/107          * VU2950027800                *  251      *  90     *         *  0      *  1      *
      18 2/C/2/107          * VU2950027300                *  260      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
      19 3/C/2/107          * VU2900027500                *  260      *  90     *         *  0      *  1      *
      20 MORT/2/107         * VU2970030300                *  220      *  90     *         *  0      *  1      *
      21 RECON/107          * VU5500023000                *   17      *  90     *         *  0      *  1      *
--------------------------------------------------------------------------------------------------------------
```

Sample Army Status Report (Continued)

```
EQUIPMENT LEVEL REPORT
  TIME OF THIS REPORT (NOW)    030302
  TIME OF LAST REPORT (LAST)
        *CHNG=NOW-LAST

        UNIT             EQUIPMENT      BASIC  LEVELS     NO. MANNED         EQUIPMENT      BASIC  LEVELS     NO. MANN
NO.        NAME        NO.    TYPE      LOAD  NOW CHNG*  NOW CHNG      NO.     TYPE       LOAD  NOW CHNG*  NOW CHNG
                       *                                          **
                       *                                          **
 1/107 (RED)           *                                          **
                       *                                          **
    1 1/A/1/107        * 2  7.62 MM LMG    6     6          6     ** 4   RPG-7            3     3          3
                       * 5  SPG-9          1     1          1     ** 6   AT-3             1     1          1
                       *10  T-62/115       5     5          5     **11   BMP/SAGGER       3     3          3
                       *18  SA-7           1     1          0     ** 1   AKM             62    62         26
                       *                                          **
    2 2/A/1/107        * 2  7.62 MM LMG    6     6          6     ** 4   RPG-7            3     3          3
                       * 9  BRDM/SAGGER    1     1          1     **11   BMP/SAGGER       3     3          3
                       *18  SA-7           1     1          0     ** 1   AKM             32    32          9
                       *                                          **
    3 3/A/1/107        * 2  7.62 MM LMG    7     7          7     ** 4   RPG-7            3     3          3
                       * 8  100MM AT GUN   1     1          1     **11   BMP/SAGGER       4     4          4
                       *18  SA-7           1     1          0     **21   SA-9             1     1          0
                       *40  TRUCK          3     3          3     ** 1   AKM             50    50         18
                       *                                          **
    4 1/B/1/107        * 2  7.62 MM LMG    6     6          6     ** 4   RPG-7            3     3          3
                       * 9  BRDM/SAGGER    1     1          1     **10   T-62/115         4     4          4
                       *11  BMP/SAGGER     3     3          3     **18   SA-7             1     1          0
                       * 1  AKM           48    48         13     **
                       *                                          **
    5 2/B/1/107        * 2  7.62 MM LMG    6     6          6     ** 4   RPG-7            3     3          3
                       * 5  SPG-9          1     1          1     ** 6   AT-3             1     1          1
                       *11  BMP/SAGGER     3     3          3     **18   SA-7             1     1          0
                       * 1  AKM           42    42         21     **
                       *                                          **
    6 3/B/1/107        * 2  7.62 MM LMG    7     7          7     ** 4   RPG-7            3     3          3
                       * 8  100MM AT GUN   1     1          1     **11   BMP/SAGGER       4     4          4
                       *18  SA-7           1     2          0     **21   SA-9             1     1          1
```

Sample Army Status Report (Continued)

```
AMMUNITION LEVEL REPORT
  TIME OF THIS REPORT (NOW)   030302
  TIME OF LAST REPORT (LAST)
       *CHNG=NOW-LAST

    UNIT              AMMUNITION        BASIC    LEVELS            AMMUNITION       BASIC    LEVELS
NO.       NAME      NO.    TYPE         LOAD   NOW    CHNG*     NO.    TYPE        LOAD    NOW   CHNG*
                    *                                          **
                    *                                          **
 1/107 (RED)        *                                          **
                    *                                          **
   1 1/A/1/107      * 1   AKM RNDS     27900  27900            ** 2   7.62MM LT    12000  12000
                    * 4   82MM AT RKT     36     36            ** 5   SPG 9 RNDS      18     18
                    * 9   SAGGER          17     17            **10   115MM SHELL    200    200
                    *18   SA-7 MISSILE    12     12            **
                    *                                          **
   2 2/A/1/107      * 1   AKM RNDS     23400  23400            ** 2   7.62MM LT    12000  12000
                    * 4   82MM AT RKT     36     36            ** 9   SAGGER          27     27
                    *18   SA-7 MISSILE    12     12            **
                    *                                          **
   3 3/A/1/107      * 1   AKM RNDS     22500  22500            ** 2   7.62MM LT    14000  14000
                    * 4   82MM AT RKT     36     36            ** 8   100MM SHELL    110    110
                    * 9   SAGGER          20     20            **11                    4      4
                    *                                          **
   4 1/B/1/107      * 1   AKM RNDS     21600  21600            ** 2   7.62MM LT    12000  12000
                    * 4   82MM AT RKT     36     36            ** 9   SAGGER          27     27
                    *10   115MM SHELL    160    160            **18   SA-7 MISSILE    12     12
                    *                                          **
   5 2/B/1/107      * 1   AKM RNDS     26100  26100            ** 2   7.62MM LT    12000  12000
                    * 4   82MM AT RKT     36     36            ** 5   SPG 9 RNDS      18     18
                    * 9   SAGGER          17     17            **18   SA-7 MISSILE    12     12
                    *                                          **
   6 3/B/1/107      * 1   AKM RNDS     21600  21600            ** 2   7.62MM LT    14000  14000
                    * 4   82MM AT RKT     36     36            ** 8   100MM SHELL    110    110
                    * 9   SAGGER          20     20            **18   SA-7 MISSILE    12     12
                    *21   SA-9 MISSILE     4      4            **
                    *                                          **
   7 1/C/1/107      * 1   AKM RNDS     18000  18000            ** 2   7.62MM LT    12000  12000
```

Sample Army Status Report (Continued)

```
        FUEL LEVEL REPORT
          TIME OF THIS REPORT (NOW)    030302
          TIME OF LAST REPORT (LAST)
                *CHNG=NOW-LAST

                           GASOLINE (GAL.)         DIESEL (GAL.)          AVIATION FUEL (LBS.)
         UNIT              BASIC    LEVELS         BASIC    LEVELS         BASIC    LEVELS
NO.          NAME          LOAD  NOW   CHNG*       LOAD  NOW   CHNG*       LOAD  NOW   CHNG*
                        *                      **                      **
                        *                      **                      **
                        *                      **                      **
  1/107 (RED)           *                      **                      **
     1 1/A/1/107        *     0     0          ** 1648  1648           **     0     0
     2 2/A/1/107        *    77    77          **  198   198           **     0     0
     3 3/A/1/107        *   165   165          **  590   590           **     0     0
     4 1/B/1/107        *    77    77          ** 1358  1358           **     0     0
     5 2/B/1/107        *     0     0          **  198   198           **     0     0
     6 3/B/1/107        *   165   165          **  590   590           **     0     0
     7 1/C/1/107        *    77    77          **  778   778           **     0     0
     8 2/C/1/107        *     0     0          **  778   778           **     0     0
     9 3/C/1/107        *     0     0          **  264   264           **     0     0
    10 MORT/1/107       *   330   330          **    0     0           **     0     0
                        *                      **                      **
  2/107 (RED)           *                      **                      **
    11 1/A/2/107        *    55    55          ** 1648  1648           **     0     0
    12 2/A/2/107        *   132   132          **  198   198           **     0     0
    13 3/A/2/107        *   165   165          **  590   590           **     0     0
    14 1/B/2/107        *    77    77          ** 1358  1358           **     0     0
    15 2/B/2/107        *     0     0          **  198   198           **     0     0
    16 3/B/2/107        *   165   165          **  590   590           **     0     0
    17 1/C/2/107        *    77    77          ** 1358  1358           **     0     0
    18 2/C/2/107        *     0     0          **  198   198           **     0     0
    19 3/C/2/107        *     0     0          **  524   524           **     0     0
    20 MORT/2/107       *   330   330          **  110   110           **     0     0
    21 RECON/107        *   264   264          **  198   198           **     0     0
                        *                      **                      **
  3/107 (RED)           *                      **                      **
    22 A/3/107          *   242   242          **  920   920           **     0     0
```

Sample Army Status Report (Continued)

```
PERSONNEL LEVEL REPORT
     TIME OF THIS REPORT (NOW)     030302
     TIME OF LAST REPORT (LAST)
          *CHNG=NOW-LAST

        UNIT              TOTAL PERSONNEL        CO             OFFICERS            EMLP              EM
NO.         NAME          TOE    NOW CHNG* TOE   NOW CHNG*  TOE   NOW CHNG*  TOE   NOW CHNG*  TOE   NOW CHNG*
                       *                 *                **                **                **
                       *                 *                **                **                **

                       *                 *                **                **                **
  1/107 (RED)          *                 *                **                **                **
    1 1/A/1/107        *   68    68      *   1    1       **   3    3       **   9    9       **  55   55
    2 2/A/1/107        *   38    38      *   1    1       **   2    2       **   4    4       **  31   31
    3 3/A/1/107        *   56    56      *   1    1       **   3    3       **   8    8       **  44   44
    4 1/B/1/107        *   54    54      *   1    1       **   2    2       **   8    8       **  43   43
    5 2/B/1/107        *   48    48      *   1    1       **   1    1       **   5    5       **  41   41
    6 3/B/1/107        *   54    54      *   1    1       **   2    2       **   8    8       **  43   43
    7 1/C/1/107        *   46    46      *   1    1       **   2    2       **   6    6       **  37   37
    8 2/C/1/107        *   44    44      *   1    1       **   1    1       **   6    6       **  36   36
    9 3/C/1/107        *   38    38      *   1    1       **   1    1       **   5    5       **  31   31
   10 MORT/1/107       *   50    50      *   1    1       **   3    3       **   4    4       **  42   42
                       *                 *                **                **                **
  2/107 (RED)          *                 *                **                **                **
   11 1/A/2/107        *   68    68      *   1    1       **   3    3       **   9    9       **  55   55
   12 2/A/2/107        *   38    38      *   1    1       **   2    2       **   4    4       **  31   31
   13 3/A/2/107        *   56    56      *   1    1       **   3    3       **   8    8       **  44   44
   14 1/B/2/107        *   54    54      *   1    1       **   2    2       **   8    8       **  43   43
   15 2/B/2/107        *   48    48      *   1    1       **   1    1       **   5    5       **  41   41
   16 3/B/2/107        *   54    54      *   1    1       **   2    2       **   8    8       **  43   43
   17 1/C/2/107        *   54    54      *   1    1       **   2    2       **   8    8       **  43   43
   18 2/C/2/107        *   36    36      *   1    1       **   1    1       **   4    4       **  30   30
   19 3/C/2/107        *   46    46      *   1    1       **   1    1       **   7    7       **  37   37
   20 MORT/2/107       *   54    54      *   1    1       **   3    3       **   5    5       **  45   45
   21 RECON/107        *   47    47      *   1    1       **   3    3       **   5    5       **  38   38
                       *                 *                **                **                **
  3/107 (RED)          *                 *                **                **                **
   22 A/3/107          *  135   135      *   1    1       **   8    8       **  16   16       ** 110  110
   23 B/3/107          *  139   139      *   1    1       **   7    7       **  16   16       ** 115  115
```

D	Unit is under maneuver command and control input via the menus and is moving dismounted.
F	Unit is under fire command and control input via the menus.

5.7.5.2 Assumptions and Data Sources

The obvious assumption here is that a printed record of the status would be useful to the controllers both during and after a game.

The source for the format of the status report and inclusion or exclusion of various numbers was a meeting in March 1976 at Fort Benning between TRW representatives; Ed Goldberg, George Wilde, and Bob Milder, and the Fort Benning CATTS Directorate personnel and Naval Training Equipment Center representatives.

5.7.5.3 Equations

None applicable to this submodule.

5.7.6 Diagnostic and Miscellaneous Output Submodule

5.7.6.1 Operation

There are other printouts available from the math model in addition to the Army status report. Mostly, these are printouts used during development of CATTS to debug the various modules. Some of this debug printout might prove useful to the controllers, and all of it would certainly be useful should problems occur or for debugging future changes. All of the debug printout available and the variables which turn the printouts on and off are shown in Appendix B of the Data Base/Operations Manual. The TRW status report is one of the printouts that was intended for controller use. It contains the same information as the Army status report plus some information that is more programmatical than tactical. A sample of the first pages of each of the 5 sections is shown on the next page.

Each of the 5 sections corresponds to the 5 sections of the Army status report (see Section 5.7.5). Each of the 5 sections of the TRW status report can be individually turned on every N minutes or off when N = 0 by namelisting IOINTRVL of 1, 2, 3, 4, and 13 equal to N in the run deck. The 1, 2, 3, 4, and 13 correspond to unit, equipment, ammunition, personnel, and fuel reports, respectively. The default in the namelist portion of the data base file is zero for all 15 IOINTRVL elements.

The other printout intended for controller use is a casualty report which is the first part of the post game summary (the other part of the post game summary being the final status report). This casualty report shows, for red then blue, which weapons destroyed which enemy equipment and personnel, and the amounts each weapon destroyed. This report is always produced if the game is normally terminated (SOP #7 in Appendix B of the Data Base/Operations Manual).

The diagnostic output is scattered throughout the other modules of CATTS. The only readily separable portions are those which produce the TRW status report (STATREP) and the casualty report (CASREP). Subroutine linkages for these are shown in Figure 5-103, while Table 5-54 gives a brief description of each with its principal inputs and outputs.

Shown on the last pages of this section is a sample of this casualty report (produced by subroutine CASREP).

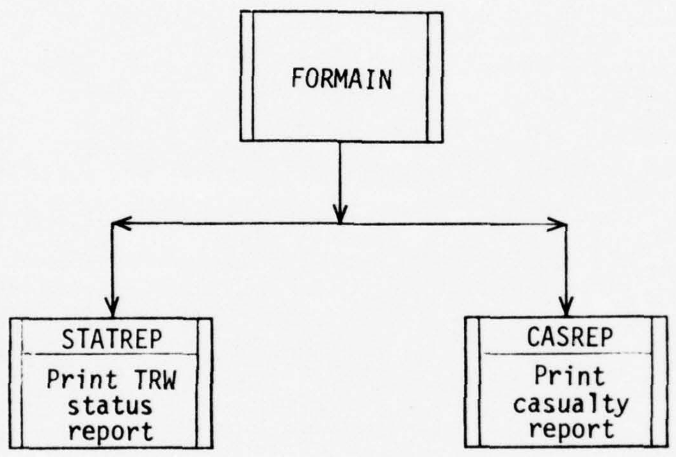


Figure 5-103. Subroutine Linkage for the Diagnostic Submodule

Table 5-54. Diagnostic and Miscellaneous Output Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
STATREP	Prints various diagnostic reports based on values of input flags. Reports are not printed asynchronously, hence can seriously degrade real-time performance.	Input: IOINTRVL - See pages 158-160 of the Data Base/Operations Manual. Output: See sample printouts in this section of the User's Manual.
CASREP	Prints a Post-Game Casualty Report Summary.	Input: NQPTOT(ICOL) - Number of different equipment types belonging to all red (ICOL = 1) and all blue (ICOL = 2) units. NDXEQ(IEQ,ICOL) - Index into STATS table for equipment type IEQ when belonging to red (ICOL = 1) or blue (ICOL = 2) units. STATS(IND1,IND2,ICOL) - Number of pieces of equipment type of index IND1 destroyed by equipment type of index IND2. ICOL = 2 is for red equipments destroyed by blue weapons, and ICOL = 1 for blue destroyed by red. This is true for 1 ≤ IND1 ≤ NQPTOT(ICOL). However, when IND1 = NQPTOT(ICOL) + 1 then casualties are accumulated for personnel killed by equipment of index IND2, while when IND1 = NQPTOT(ICOL) + 2 equipment failures of equipment of index IND2 are accumulated. Example: Suppose blue equipment I kills one red equip-

Table 5-54. Diagnostic and Miscellaneous Output Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
CASREP (Cont'd)		ment J. Then N1 = NDXEQ(I,2) N2 = NDXEQ(J,1) STATS(N1, N2, 2) = STATS(N1, N2, 2)+ 1 Output: See example of Casualty Report in this section.

Sample TRW Status Report

```
                       UNIT STATUS REPORT AT TIME    3: 3: 0
                               CC      X      Y      Z ANGLE  MOVE ONOFF DC/OP TRAVL  MOVE   MOVE   MOVE   MOVE  FCT  RED  SLP  VEG SOIL
                              FLG  COORD  COORD  COORD FACED  RATE  ROAD STATE  CODE  CODE  DATA1  DATA2  DATA3  SUP  CON  PCT  TYP  TYP

OPGP  1 1/107 (RED)                30100  29957           90                3     2     7      0      0      0
UNIT  1 1/A/1/107                  29800  30300    220    90     0   OFF   22     2     7      0      0      0    0    0    0    1    3
UNIT  2 2/A/1/107                  29800  30900    213    90     0   OFF   22     2     7      0      0      0    0    0    1    1    3
UNIT  3 3/A/1/107                  29200  30500    234    90     0   OFF   22     2     7      0      0      0    0    0    0    1    3
UNIT  4 1/B/1/107                  29000  29600    210    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT  5 2/B/1/107                  29000  30200    210    90     0   OFF   22     2     7      0      0      0    0    0   10    1    3
UNIT  6 3/B/1/107                  28500  30000    212    90     0   OFF   22     2     7      0      0      0    0    0    1    1    3
UNIT  7 1/C/1/107                  29000  30900    228    90     0   OFF   22     2     7      0      0      0    0    0   -6    1    3
UNIT  8 2/C/1/107                  29000  31600    225    90     0   OFF   22     2     7      0      0      0    0    0    3    1    3
UNIT  9 3/C/1/107                  28500  31200    227    90     0   OFF   22     2     7      0      0      0    0    0    0    1    1
UNIT 10 MORT/1/107                 29800  31700    225    90     0   OFF   69     2     7      0      0      0    0    0    3    1    3

OPGP  2 2/107 (RED)                29700  29471           90                3     2     7      0      0      0
UNIT 11 1/A/2/107                  30100  29000    225    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT 12 2/A/2/107                  30100  28500    232    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT 13 3/A/2/107                  29600  28800    230    90     0   OFF   22     2     7      0      0      0    0    0   -2    1    3
UNIT 14 1/B/2/107                  29100  29000    223    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT 15 2/B/2/107                  29100  28400    241    90     0   OFF   22     2     7      0      0      0    0    0   -3    1    3
UNIT 16 3/B/2/107                  28500  28800    232    90     0   OFF   22     2     7      0      0      0    0    0   -4    1    3
UNIT 17 1/C/2/107                  29500  27800    251    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT 18 2/C/2/107                  29500  27300    260    90     0   OFF   22     2     7      0      0      0    0    0   -2    1    3
UNIT 19 3/C/2/107                  29000  27500    260    90     0   OFF   22     2     7      0      0      0    0    0   -2    1    3
UNIT 20 MORT/2/107                 29700  30300    220    90     0   OFF   69     2     7      0      0      0    0    0    0    1    3

UNIT 21 RECON/107                  55000  23000     17    90     0   OFF   22     2     7      0      0      0    0    0    0    1    3

OPGP  3 3/107 (RED)                38500  28889           90                3     2     7      0      0      0
UNIT 22 A/3/107                    35500  28700    145    90     0   OFF   22     2     7      0      0      0    0    0    0    1    1
UNIT 23 B/3/107                    33900  29000    165    90     0   OFF   22     2     7      0      0      0    0    0   -1    1    3
UNIT 24 C/3/107                    37000  28000    134    90     0   OFF   22     2     7      0      0      0    0    0    0    1    1
UNIT 25 MORT/3/107                 33000  29400    165    90     0   OFF   69     2     7      0      0      0    0    0    0    1    3
UNIT 26 C/1A/107                   38500  27300    122    90     0   OFF   22     2     7      0      0      0    0    0    0    1    1

UNIT 27 HOW BTRY/107               51000  20000     27    90     0   OFF   69     2     7      0      0      0    0    0    0    1    3
UNIT 28 HOW BN/CAA                 51000  22000     34    90     0   OFF   69     2     7      0      0      0    0    0    0    1    3
```

Sample TRW Status Report (Continued)

```
                  EQUIPMENT LEVEL REPORT AT TIME    3: 3: 0
  UNIT         **     EQUIPMENT    BASIC/ CURR/NUMBR **     EQUIPMENT    BASIC/ CURR/NUMBR **     EQUIPMENT      BASIC/ CURR/NUMBR
# NAME              # NAME          LOAD/LEVEL/MANND **   # NAME          LOAD/LEVEL/MANND **   # NAME            LOAD/LEVEL/MANND
1 1/A/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  5 SPG-9              1/    1/    1
               **   6 AT-3            1/    1/    1 ** 10 T-62/115        5/    5/    5 ** 11 BMP/SAGGER         3/    3/    3
               **  18 SA-7            1/    1/    0 **  1 AKM            62/   62/   26 **
2 2/A/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  9 BRDM/SAGGER        1/    1/    1
               **  11 BMP/SAGGER      3/    3/    3 ** 18 SA-7            1/    1/    0 **  1 AKM               32/   32/    9
3 3/A/1/107    **   2 7.62 MM LMG     7/    7/    7 **  4 RPG-7           3/    3/    3 **  8 100MM AT GUN       1/    1/    1
               **  11 BMP/SAGGER      4/    4/    4 ** 18 SA-7            1/    1/    0 ** 21 SA-9               1/    1/    0
               **  40 TRUCK           3/    3/    3 **  1 AKM            50/   50/   18 **
4 1/B/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  9 BRDM/SAGGER        1/    1/    1
               **  10 T-62/115        4/    4/    4 ** 11 BMP/SAGGER      3/    3/    3 ** 18 SA-7               1/    1/    0
               **   1 AKM            48/   48/   13 **
5 2/B/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  5 SPG-9              1/    1/    1
               **   6 AT-3            1/    1/    1 ** 11 BMP/SAGGER      3/    3/    3 ** 18 SA-7               1/    1/    0
               **   1 AKM            42/   42/   21 **
6 3/B/1/107    **   2 7.62 MM LMG     7/    7/    7 **  4 RPG-7           3/    3/    3 **  8 100MM AT GUN       1/    1/    1
               **  11 BMP/SAGGER      4/    4/    4 ** 18 SA-7            1/    2/    0 ** 21 SA-9               1/    1/    1
               **  40 TRUCK           3/    3/    3 **  1 AKM            48/   48/   12 **
7 1/C/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  9 BRDM/SAGGER        1/    1/    1
               **  10 T-62/115        2/    2/    2 ** 11 BMP/SAGGER      3/    3/    3 ** 18 SA-7               1/    1/    0
               **   1 AKM            40/   40/   11 **
8 2/C/1/107    **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 ** 10 T-62/115           2/    2/    2
               **  11 BMP/SAGGER      3/    3/    3 ** 18 SA-7            2/    2/    0 **  1 AKM               38/   38/   11
9 3/C/1/107    **   2 7.62 MM LMG     7/    7/    7 **  4 RPG-7           3/    3/    3 ** 11 BMP/SAGGER         4/    4/    4
               **  18 SA-7            2/    2/    0 **  1 AKM            32/   32/    6 **
10 MORT/1/107  **  13 120 MM MORT     6/    6/    6 ** 40 TRUCK           6/    6/    6 **  1 AKM               52/   52/    8
11 1/A/2/107   **   2 7.62 MM LMG     7/    7/    7 **  4 RPG-7           3/    3/    3 **  5 SPG-9              1/    1/    1
               **   6 AT-3            1/    1/    1 ** 10 T-62/115        5/    5/    5 ** 11 BMP/SAGGER         3/    3/    3
               **  18 SA-7            1/    1/    0 **  1 AKM            62/   62/   23 ** 40 TRUCK              1/    1/    1
12 2/A/2/107   **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  9 BRDM/SAGGER        1/    1/    1
               **  11 BMP/SAGGER      3/    3/    3 ** 18 SA-7            1/    1/    0 **  1 AKM               32/   28/    8
               **  40 TRUCK           1/    1/    1 **
13 3/A/2/107   **   2 7.62 MM LMG     7/    7/    7 **  4 RPG-7           3/    3/    3 **  8 100MM AT GUN       1/    1/    1
               **  11 BMP/SAGGER      4/    4/    4 ** 18 SA-7            1/    1/    0 ** 21 SA-9               1/    1/    0
               **  40 TRUCK           3/    3/    3 **  1 AKM            50/   50/   18 **
14 1/B/2/107   **   2 7.62 MM LMG     6/    6/    6 **  4 RPG-7           3/    3/    3 **  9 BRDM/SAGGER        1/    1/    1
               **  10 T-62/115        4/    4/    4 ** 11 BMP/SAGGER      3/    3/    3 ** 18 SA-7               1/    1/    0
               **   1 AKM            48/   48/   13 **
```

Sample TRW Status Report (Continued)

```
                       AMMO LEVEL REPORT AT TIME   3: 3: 0
    UNIT            **    AMMO              BASIC/ CURRENT **    AMMO              BASIC/ CURRENT **    AMMO               BASIC/ CURRENT
  # NAME            **  # NAME               LOAD/   LEVEL **  # NAME               LOAD/   LEVEL **  # NAME                LOAD/   LEVEL
  1 1/A/1/107       **  1 AKM RNDS          27900/   27900 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  5 SPG 9 RNDS           18/      18 **  9 SAGGER               17/      17 ** 10 115MM SHELL          200/     200
                    ** 18 SA-7 MISSILE         12/      12 **
  2 2/A/1/107       **  1 AKM RNDS          23400/   23400 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 18 SA-7 MISSILE         12/      12 **
  3 3/A/1/107       **  1 AKM RNDS          22500/   22500 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  8 100MM SHELL         110/     110 **  9 SAGGER               20/      20 ** 11                        4/       4
  4 1/B/1/107       **  1 AKM RNDS          21600/   21600 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 10 115MM SHELL         160/     160 ** 18 SA-7 MISSILE          12/      12
  5 2/B/1/107       **  1 AKM RNDS          26100/   26100 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  5 SPG 9 RNDS           18/      18 **  9 SAGGER               17/      17 ** 18 SA-7 MISSILE          12/      12
  6 3/B/1/107       **  1 AKM RNDS          21600/   21600 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  8 100MM SHELL         110/     110 **  9 SAGGER               20/      20 ** 18 SA-7 MISSILE          12/      12
                    ** 21 SA-9 MISSILE          4/       4 **
  7 1/C/1/107       **  1 AKM RNDS          18000/   18000 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 10 115MM SHELL          80/      80 ** 18 SA-7 MISSILE          12/      12
  8 2/C/1/107       **  1 AKM RNDS          17100/   17100 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               15/      15 ** 10 115MM SHELL          80/      80 ** 18 SA-7 MISSILE          24/      24
  9 3/C/1/107       **  1 AKM RNDS          14400/   14400 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               20/      20 ** 18 SA-7 MISSILE         24/      24 **
 10 MORT/1/107      ** 13 120MM MORT         1080/    1080 **  1 AKM RNDS          22500/   22500 **
 11 1/A/2/107       **  1 AKM RNDS          27900/   27900 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  5 SPG 9 RNDS           18/      18 **  9 SAGGER               17/      17 ** 10 115MM SHELL          200/     200
                    ** 18 SA-7 MISSILE         12/      12 **
 12 2/A/2/107       **  1 AKM RNDS          23400/   23400 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 18 SA-7 MISSILE         12/      12 **
 13 3/A/2/107       **  1 AKM RNDS          22500/   22500 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  8 100MM SHELL         110/     110 **  9 SAGGER               20/      20 ** 18 SA-7 MISSILE          12/      12
 14 1/B/2/107       **  1 AKM RNDS          21600/   21600 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 10 115MM SHELL         160/     160 ** 18 SA-7 MISSILE          12/      12
 15 2/B/2/107       **  1 AKM RNDS          26100/   26100 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  5 SPG 9 RNDS           18/      18 **  9 SAGGER               17/      17 ** 18 SA-7 MISSILE          12/      12
 16 3/B/2/107       **  1 AKM RNDS          21600/   21600 **  2 7.62MM LT         14000/   14000 **  4 82MM AT RKT           36/      36
                    **  8 100MM SHELL         110/     110 **  9 SAGGER               20/      20 ** 18 SA-7 MISSILE          12/      12
                    ** 21 SA-9 MISSILE          4/       4 **
 17 1/C/2/107       **  1 AKM RNDS          21600/   21600 **  2 7.62MM LT         12000/   12000 **  4 82MM AT RKT           36/      36
                    **  9 SAGGER               27/      27 ** 10 115MM SHELL         160/     160 ** 18 SA-7 MISSILE          12/      12
```

Sample TRW Status Report (Continued)

```
                          FUEL LEVEL REPORT AT TIME    3: 3: 0
   UNIT            **   GAS  BASIC/CURRENT** DIESEL BASIC/CURRENT**  AVGAS BASIC/CURRENT
 # NAME            **   GALS  LOAD/  LOAD **   GALS  LOAD/  LOAD **   GALS  LOAD/  LOAD
 1 1/A/1/107       **            0/     0 **         1648/  1648 **            0/     0
 2 2/A/1/107       **           77/    77 **          198/   198 **            0/     0
 3 3/A/1/107       **          165/   165 **          590/   590 **            0/     0
 4 1/B/1/107       **           77/    77 **         1358/  1358 **            0/     0
 5 2/B/1/107       **            0/     0 **          198/   198 **            0/     0
 6 3/B/1/107       **          165/   165 **          590/   590 **            0/     0
 7 1/C/1/107       **           77/    77 **          778/   778 **            0/     0
 8 2/C/1/107       **            0/     0 **          778/   778 **            0/     0
 9 3/C/1/107       **            0/     0 **          264/   264 **            0/     0
10 MORT/1/107      **          330/   330 **            0/     0 **            0/     0
11 1/A/2/107       **           55/    55 **         1648/  1648 **            0/     0
12 2/A/2/107       **          132/   132 **          198/   198 **            0/     0
13 3/A/2/107       **          165/   165 **          590/   590 **            0/     0
14 1/B/2/107       **           77/    77 **         1358/  1358 **            0/     0
15 2/B/2/107       **            0/     0 **          198/   198 **            0/     0
16 3/B/2/107       **          165/   165 **          590/   590 **            0/     0
17 1/C/2/107       **           77/    77 **         1358/  1358 **            0/     0
18 2/C/2/107       **            0/     0 **          198/   198 **            0/     0
19 3/C/2/107       **            0/     0 **          524/   524 **            0/     0
20 MORT/2/107      **          330/   330 **          110/   110 **            0/     0
21 RECON/107       **          264/   264 **          198/   198 **            0/     0
22 A/3/107         **          242/   242 **          920/   920 **            0/     0
23 B/3/107         **          242/   242 **          920/   920 **            0/     0
24 C/3/107         **           77/    77 **         1030/  1030 **            0/     0
25 MORT/3/107      **          330/   330 **          110/   110 **            0/     0
26 C/TK/107        **            0/     0 **         4060/  4060 **            0/     0
27 HOW BTRY/107    **          330/   330 **          220/   220 **            0/     0
28 HOW BN/CAA      **          990/   990 **          370/   370 **            0/     0
29 HOW BN/17F      **          990/   990 **            0/     0 **            0/     0
30 GH BN/CAA       **          990/   990 **          110/   110 **            0/     0
31 SAM 17F/CAA     **          220/   220 **          528/   528 **            0/     0
32 A/1/79          **           55/    55 **         3300/  3300 **            0/     0
33 B/1/79          **           55/    55 **         3160/  3160 **            0/     0
34 C/1/79          **           55/    55 **         3160/  3160 **            0/     0
39 BN CP           **          550/   550 **         1507/  1507 **            0/     0
47 CD GP/A/2-77    **           55/    55 **          137/   137 **            0/     0
49 1/A/2-77        **            0/     0 **          548/   548 **            0/     0
```

Sample TRW Status Report (Continued)

```
                 PERSONNEL LEVEL REPORT AT TIME   3: 3: 0
   UNIT            TOTAL MEN    *      CO        *  OFFICERS    *    EMLP       *    EM      , *** MEN PER VULNERABILITY CLASS
 # NAME            TOE  CURR *   TOE   CURR *   TOE  CURR *   TOE  CURR *   TOE  CURR ***CLASS 1     2     3     4
 1 1/A/1/107        68    68 *     1      1 *     3     3 *     9     9 *    55    55 ***     68     0     0     0
 2 2/A/1/107        38    38 *     1      1 *     2     2 *     4     4 *    31    31 ***     38     0     0     0
 3 3/A/1/107        56    56 *     1      1 *     3     3 *     8     8 *    44    44 ***     56     0     0     0
 4 1/B/1/107        54    54 *     1      1 *     2     2 *     8     8 *    43    43 ***     54     0     0     0
 5 2/B/1/107        48    48 *     1      1 *     1     1 *     5     5 *    41    41 ***     48     0     0     0
 6 3/B/1/107        54    54 *     1      1 *     2     2 *     8     8 *    43    43 ***     54     0     0     0
 7 1/C/1/107        46    46 *     1      1 *     2     2 *     6     6 *    37    37 ***     46     0     0     0
 8 2/C/1/107        44    44 *     1      1 *     1     1 *     6     6 *    36    36 ***     44     0     0     0
 9 3/C/1/107        38    38 *     1      1 *     1     1 *     5     5 *    31    31 ***     38     0     0     0
10 MORT/1/107       50    50 *     1      1 *     3     3 *     4     4 *    42    42 ***     50     0     0     0
11 1/A/2/107        68    68 *     1      1 *     3     3 *     9     9 *    55    55 ***     68     0     0     0
12 2/A/2/107        38    38 *     1      1 *     2     2 *     4     4 *    31    31 ***     38     0     0     0
13 3/A/2/107        56    56 *     1      1 *     3     3 *     8     8 *    44    44 ***     56     0     0     0
14 1/B/2/107        54    54 *     1      1 *     2     2 *     8     8 *    43    43 ***     54     0     0     0
15 2/B/2/107        48    48 *     1      1 *     1     1 *     5     5 *    41    41 ***     48     0     0     0
16 3/B/2/107        54    54 *     1      1 *     2     2 *     8     8 *    43    43 ***     54     0     0     0
17 1/C/2/107        54    54 *     1      1 *     2     2 *     8     8 *    43    43 ***     54     0     0     0
18 2/C/2/107        36    36 *     1      1 *     1     1 *     4     4 *    30    30 ***     36     0     0     0
19 3/C/2/107        46    46 *     1      1 *     1     1 *     7     7 *    37    37 ***     46     0     0     0
20 MORT/2/107       54    54 *     1      1 *     3     3 *     5     5 *    45    45 ***     54     0     0     0
21 RECON/107        47    47 *     1      1 *     3     3 *     5     5 *    38    38 ***     47     0     0     0
22 A/3/107         135   135 *     1      1 *     8     8 *    16    16 *   110   110 ***    135     0     0     0
23 B/3/107         139   139 *     1      1 *     7     7 *    16    16 *   115   115 ***    139     0     0     0
24 C/3/107         119   119 *     1      1 *     5     5 *    15    15 *    98    98 ***    119     0     0     0
25 MORT/3/107       54    54 *     1      1 *     3     3 *     5     5 *    45    45 ***     54     0     0     0
26 C/TK/107         55    55 *     1      1 *     2     2 *    10    10 *    42    42 ***     55     0     0     0
27 HOW BTRY/107     80    80 *     1      1 *     6     6 *    10    10 *    63    63 ***     80     0     0     0
28 HOW BN/CAA      138   138 *     1      1 *     9     9 *    42    42 *    86    86 ***    138     0     0     0
29 HOW BN/17F      133   133 *     1      1 *     8     8 *    41    41 *    83    83 ***    133     0     0     0
30 GH HN/CAA       111   111 *     1      1 *     5     5 *    22    22 *    83    83 ***    111     0     0     0
31 SAM 17F/CAA      42    42 *     1      1 *     4     4 *     5     5 *    32    32 ***     42     0     0     0
32 A/1/79           45    45 *     1      1 *     3     3 *    12    12 *    29    29 ***     45     0     0     0
33 B/1/79           41    41 *     1      1 *     3     3 *    11    11 *    26    26 ***     41     0     0     0
34 C/1/79           41    41 *     1      1 *     3     3 *    11    11 *    26    26 ***     41     0     0     0
39 BN CP            61    61 *     1      1 *    13    13 *     5     5 *    42    42 ***     61     0     0     0
47 CD GP/A/2-77      7     7 *     1      1 *     1     1 *     0     0 *     5     5 ***      7     0     0     0
49 1/A/2-77         40    40 *     1      1 *     0     0 *     9     9 *    30    30 ***     40     0     0     0
```

Sample CASREP Report

```
         RED PERSONNEL AND EQUIPMENT LOSSES BY WEAPON TYPE AT TIME     88

  WEAPON                                        EQUIPMENT TYPE
                PERSONNEL      T-62/115          BMP/SAGGER      SA-7          AKM
M-60/105 MM       14.0           2.0               .0             .0            .0
TOW               15.0           2.0              1.0             .0            .0
155 MM HOW         2.0            .0               .0             .0            .0
OTHER LOSSES        .0           1.0               .0            1.0         136.0
                --------       --------          --------      --------      --------
TOTALS            31.0           5.0              1.0            1.0         136.0


                BRDM/SAGGER    100MM AT GUN      57 MM DUAL
M-60/105 MM        2.0            .0               .0
TOW                 .0           1.0               .0
155 MM HOW          .0            .0               .0
OTHER LOSSES        .0            .0              2.0
                --------       --------          --------
TOTALS             2.0           1.0              2.0
```

Sample CASREP Report (Continued)

```
        BLUE PERSONNEL AND EQUIPMENT LOSSES BY WEAPON TYPE AT TIME      88

  WEAPON                                        EQUIPMENT TYPE
                 PERSONNEL        M-60/105 MM    AN/PPS-5A        DRAGON
BMP/SAGGER          4.0              2.0            .0              .0
122 MM HOW         35.0               .0           2.0             2.0
AIR LOSSES          5.0               .0            .0              .0
                 --------         --------       --------        --------
TOTALS             44.0              2.0           2.0             2.0
```

5.7.6.2 Assumptions and Data Sources

None applicable.

5.7.6.3 Equations

None applicable.

5.8 AIR MODULE

The CATTS Air Module is subordinate to the ground combat models, its sole purpose being to provide greater realism to the ground combat simulation. Thus, only those features of aircraft and air missions, which are necessary to provide a credible interaction with ground combat are modeled. Air units and air missions exist only for the duration of the mission. Once all the aircraft in a unit are lost, or the remaining aircraft complete the mission successfully and land, the unit ceases to exist. Ground based activities and support requirements for air units are not modeled.

5.8.1 Operation

Each model timestep, the CATTS Air Module performs all calculations necessary to the function of air units within CATTS. Location, direction, speed, and fuel consumption of air units is updated. All air-ground interactions are calculated, including ground-to-air and air-to-ground detections, air defense fire, the delivery of air weapons against ground targets, attrition to air unit from air defense, damage and casualties to roads, bridges, and ground units from air-delivered weapons, and attrition to air units resulting from support fire interference-whether advertent or inadvertent.

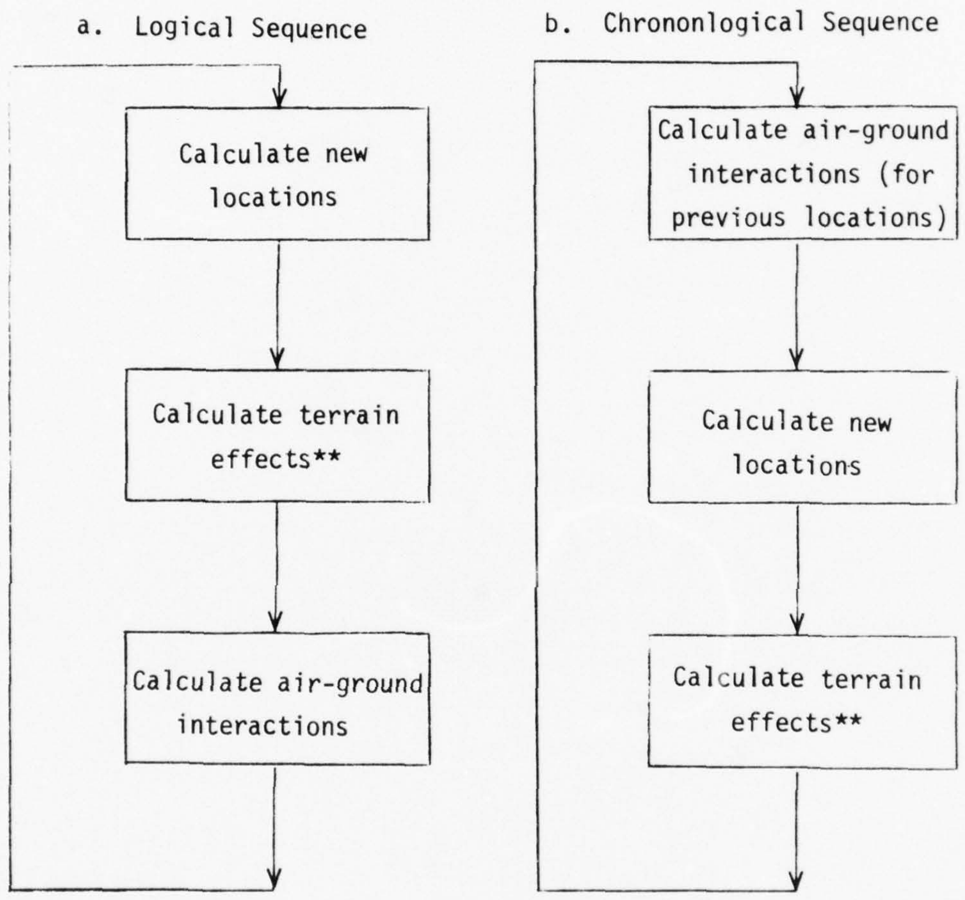
CATTS is a time-step model, with time-steps normally 60 seconds long. Ground-ground interactions occur only at the ends of time-step movement cycles. The high speed of modern tactical aircraft makes this time step too long to be practical for air units, since a fast aircraft can cover up to 20 kilometers in a single minute-long time-step. Calculating air-ground interactions only for points 60 seconds air travel time distant from each other results in absurd effects. For example, an air unit would frequently be out of range of air defense weapons in a ground unit even though it overflew the unit at some point during the 60 seconds. Thus, the high speed of aircraft relative to ground units dictated the use of a shorter time-step for the air units in order to achieve greater tactical realism. A maximum time-step of 15 seconds for air units was, therefore, called out in the CATTS specification.

The problem of integrating a 15-second time-step air model and a 60-second time-step ground model was solved in a fairly simple way which yet provides great flexibility. For each ground time-step, at least four 15-second (or less) air time-steps are calculated. Locations, speeds, altitudes, weather effects, and fuel consumption are pre-calculated for four (or more) points per air unit during each 60-second time-step. Then, during the next 60-second time-step, the air-ground interactions resulting from these four (or more) locations are calculated. Shortly after the interaction calculations, four (or more) new locations are calculated, and the cycle continues.

Calculating the air unit locations in one time-step and not calculating interactions until the next time-step seems to be unnecessary complication. However, terrain effects on air units, which are not modeled in the current CATTS, were originally intended for inclusion in the model. Computer storage and execution speed requirements dictated that all terrain effects be calculated simultaneously.

Since the terrain effects calculations occur "between" math model time-steps, it was necessary to pre-calculate locations, store them for access by the terrain model, and calculate air-ground interactions during the next time-step, after terrain effects had been calculated. This unfortunately results in a situation depicted in Figure 5-104, in which logically speaking, locations are calculated, terrain effects (could be) calculated, and then air-ground interactions are calculated; while chronologically speaking, air-ground interactions (for locations calculated during previous time-step) are calculated, then new locations, and then terrain effects for the new locations.

One more fairly straightforward complication for air module processing results from the fact that to display air unit locations as they are calculated, and before air-ground interactions are calculated, can result in misleading and incorrect displays. Air defense fire, for example, can destroy an air unit at some point during the time-step, in which case it might never reach the location displayed. For this reason, the displays of air units reflect the locations of the unit during the preceding time-step, for which air-ground interactions are already known.


a. Logical Sequence
Calculate new locations
Calculate terrain effects**
Calculate air-ground interactions
b. Chrononlogical Sequence
Calculate air-ground interactions (for previous locations)
Calculate new locations
Calculate terrain effects**

** No terrain effects actually calculated in present model

Figure 5-104. Processing Sequence for Air Module

The interactions of the CATTS air module with the other CATTS modules is depicted in Figure 5-105. Observe that the only communication with other modules is through the data base, except for calls to subroutines GRNDAIR and AIRGRND. In fact, the three chains of the air module represented by AIREVENT, AIRMOV, and AIRMOV2 communicate with each other only through the data base.

The CATTS air movement model moves airborne air units by discrete time sub-steps of not longer than 15 seconds. Each air unit has an input flight plan which specifies its route as a series of straight line segments (legs) between air control points (ACPs) as depicted in Figure 5-106. Associated with each leg are a desired altitude and velocity. The unit's altitude and velocity are incremented or decremented each time sub-step by constant amounts (which are limited by aircraft performance) until they reach the desired values for the current flight leg. This process is depicted in Figures 5-107 and 5-108. They then remain constant until the next control point is reached, at which time the process of velocity and altitude adjustment begins again. At the end of each time sub-step, local weather conditions are assessed for the feasibility of continuing the mission. Should the weather be sufficiently unfavorable, an alert message is issued and the air mission is aborted.

Each time sub-step, subroutines of the target acquisition model are called to compute air-to-ground and ground-to-air detection verdicts. Each detection results in an alert message. It also allows the detecting unit to engage the detected unit if engagement is consistent with the detecting unit's mission and within the capability of its weapons. If an engagement occurs, the damage to the engaged unit is determined by the weapons effects functions. Engagement of a ground unit by an air unit may force engagement of the air unit by the ground unit, and vice versa.

However, air units will not fire at a ground unit other than an assigned target. The controller has the capability using the interactive command and control system, of changing the mission of a unit while it is in flight. He can assign a new target, for example, on receipt of an alert message indicating that the air unit has detected a ground unit which is, for some reason, an attractive target.

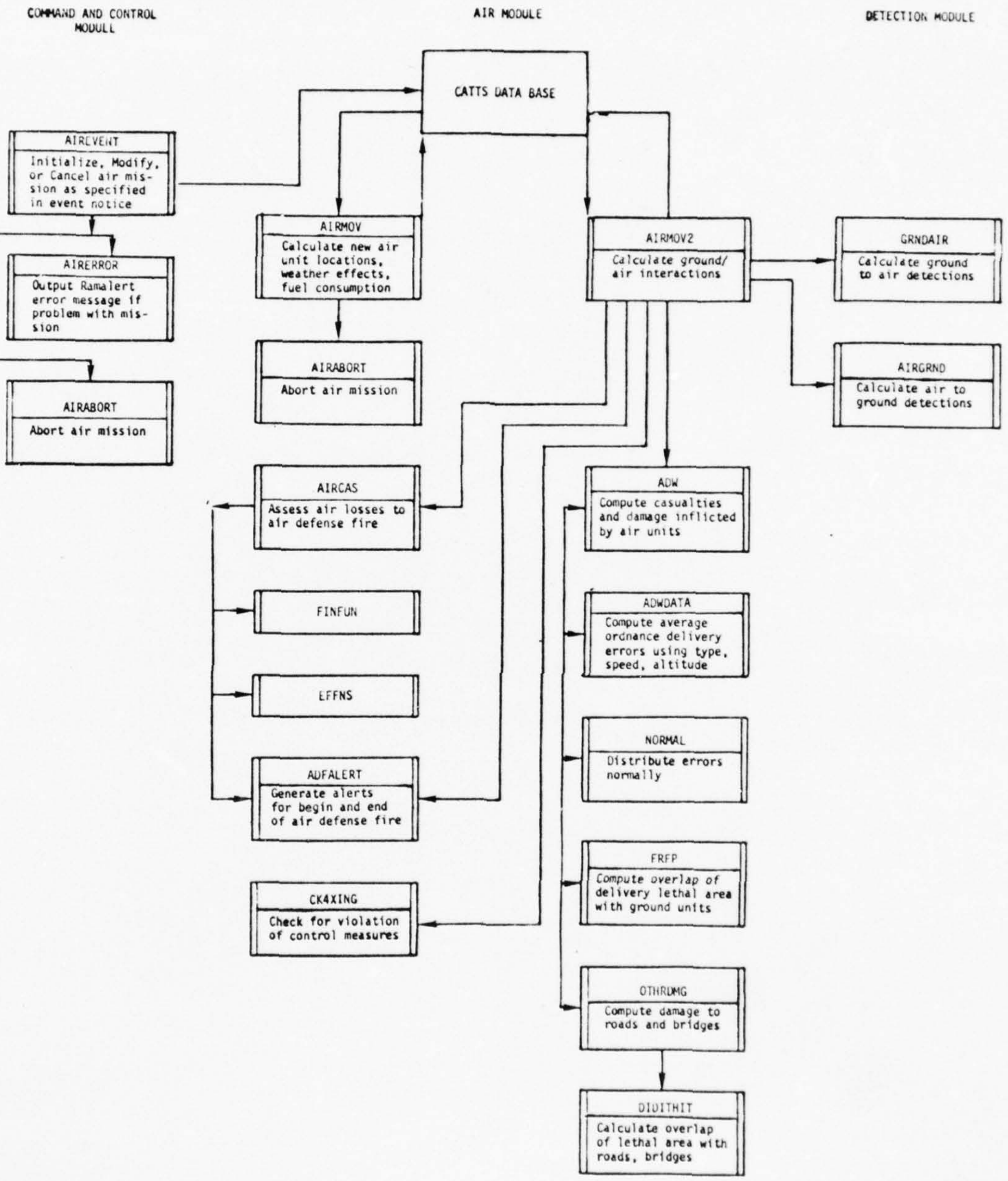
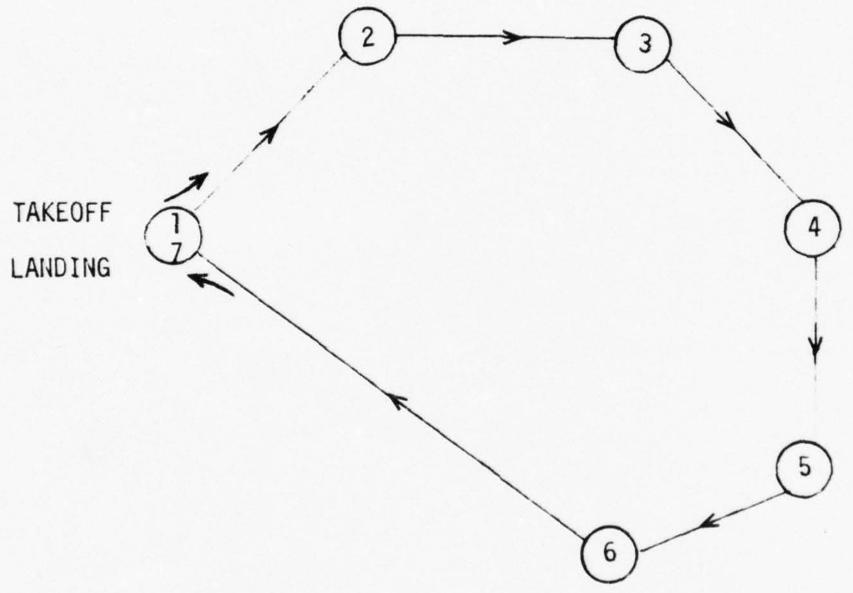

COMMAND AND CONTROL MODULE
AIR MODULE
DETECTION MODULE
CATTS DATA BASE
AIREVENT
Initialize, Modify, or Cancel air mission as specified in event notice
AIRERROR
Output Ramalert error message if problem with mission
AIRABORT
Abort air mission
AIRMOV
Calculate new air unit locations, weather effects, fuel consumption
AIRABORT
Abort air mission
AIRMOV2
Calculate ground/air interactions
GRNDAIR
Calculate ground to air detections
AIRGRND
Calculate air to ground detections
AIRCAS
Assess air losses to air defense fire
FINFUN
EFFNS
ADFALERT
Generate alerts for begin and end of air defense fire
CK4XING
Check for violation of control measures
ADW
Compute casualties and damage inflicted by air units
ADWDATA
Compute average ordnance delivery errors using type, speed, altitude
NORMAL
Distribute errors normally
FRFP
Compute overlap of delivery lethal area with ground units
OTHRDMG
Compute damage to roads and bridges
DIDITHIT
Calculate overlap of lethal area with roads, bridges

Figure 5-105. Interactions Between Air Module and Other Modules



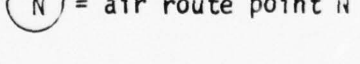


Figure 5-106. Sample Air Unit Flight Plan

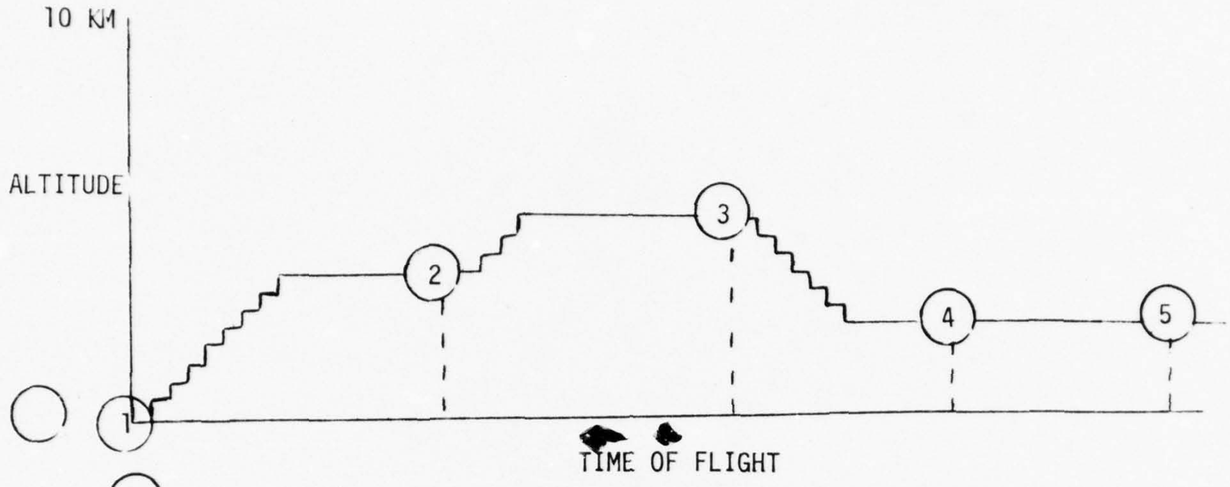


Figure 5-107. Sample Flight Altitude Contour

(N) = Air route point N

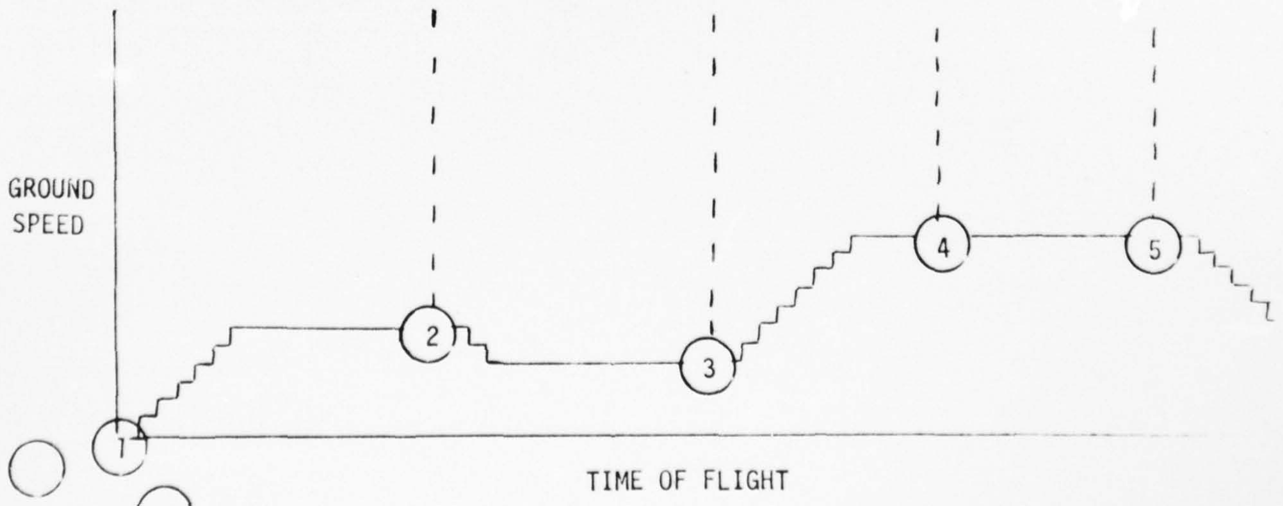


Figure 5-108. Sample Ground Speed Contour

When an air unit reaches the last point on its flight route, the landing point (which is the last checkpoint, IACPX, IACPY), a delay time is computed to model the delay time associated with the landing time delay (IAMTE(IU,2)) for the particular type of aircraft. When the delay time has elapsed, the air unit will become inactive. Note that the pre-scheduling feature of the interactive com and and control sub-system makes it possible for the input to have been made at any time previous to its desired execution.

When a new air mission is initiated, that mission is examined to see if it is within the capabilities of the aircraft. Among the factors considered are aircraft range (EQCAPAC(IEQ)) and gross payload (ROME(IEQ,1)) The latter is a function of the current weather conditions including temperature. Unfeasible conditions result in an alert message and mission cancellation. A feasible mission is initiated after appropriate delays.

The Air Module is designed to perform three general functions in the CATTS model, and within those three general functions, a number of specific functions. First, the module processes the air event notice to ensure that no requirements in the event notice are beyond the performance capabilities of the aircraft specified. Second, the module updates the location, direction, and speed of each air unit each time-step at subintervals of 1/4 minute or less. Third, the module computes all air-ground interactions for all air units and all ground units at each subinterval in the time-step. These interactions include detection, firing and casualty assessment.

The Air Module subroutine linkages are illustrated in Figures 5-109, 5-110, and 5-111, with a brief statement describing the function of each subroutine. There are three distinct chains of processing: (1) the AIREVENT chain for air event notice processing; (2) the AIRMOV chain for air unit position updating; and (3) the AIRMOV2 chain for ground-air interaction.

The Air Module is called at two different points during the execution of the mathematical model each minute. When an air mission is created, the air events processor is called to process the event notice (see Section 5.8.1.1). At its conclusion, it has stored the air unit data in appropriate common blocks for use by the remainder of the Air Module.

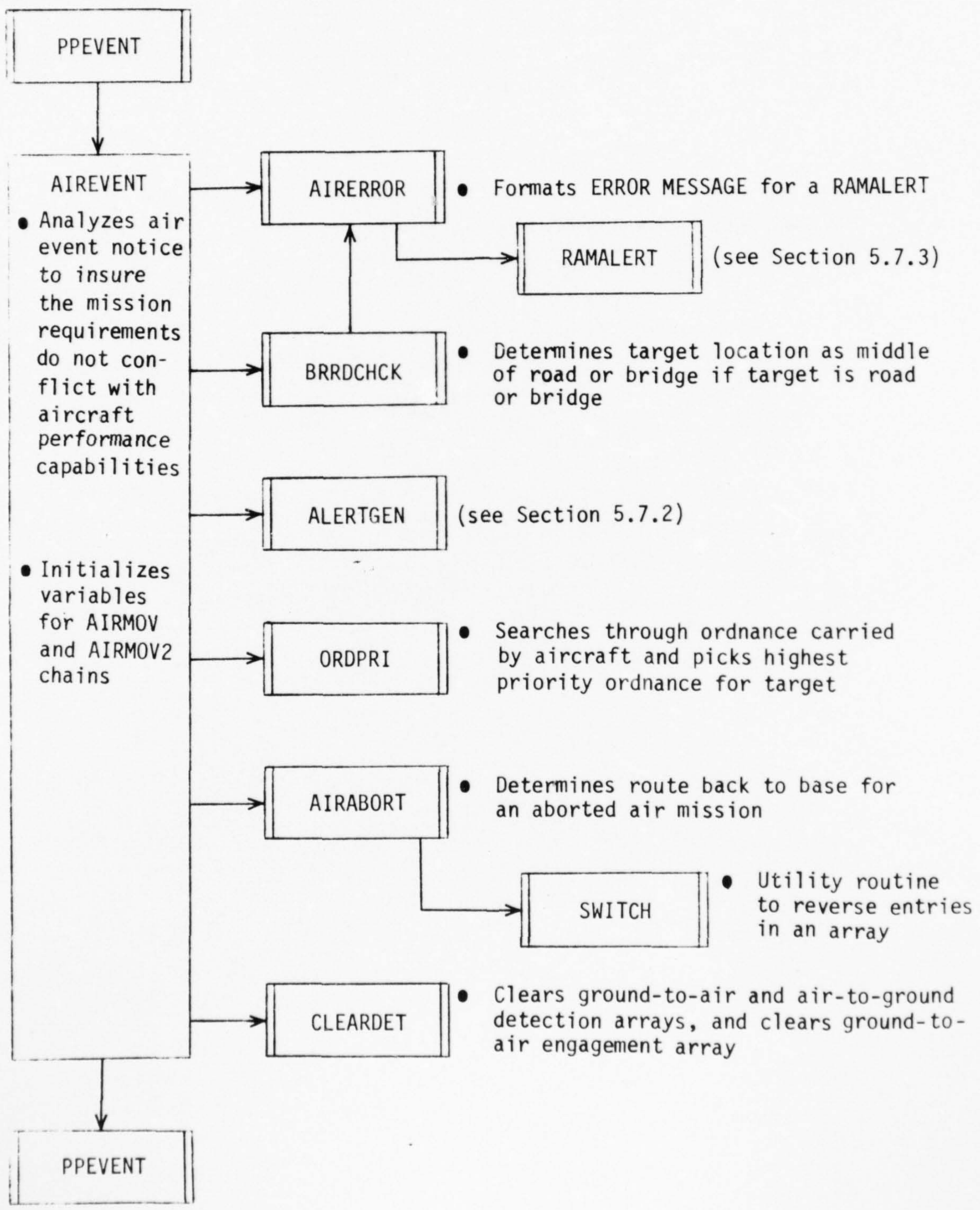

PPEVENT
AIREVENT
• Analyzes air event notice to insure the mission requirements do not conflict with aircraft performance capabilities
• Initializes variables for AIRMOV and AIRMOV2 chains
AIRERROR
• Formats ERROR MESSAGE for a RAMALERT
RAMALERT
(see Section 5.7.3)
BRRDCHCK
• Determines target location as middle of road or bridge if target is road or bridge
ALERTGEN
(see Section 5.7.2)
ORDPRI
• Searches through ordnance carried by aircraft and picks highest priority ordnance for target
AIRABORT
• Determines route back to base for an aborted air mission
SWITCH
• Utility routine to reverse entries in an array
CLEARDET
• Clears ground-to-air and air-to-ground detection arrays, and clears ground-to-air engagement array
PPEVENT

Figure 5-109. Air Module Subroutine Linkage for AIREVENT Chain

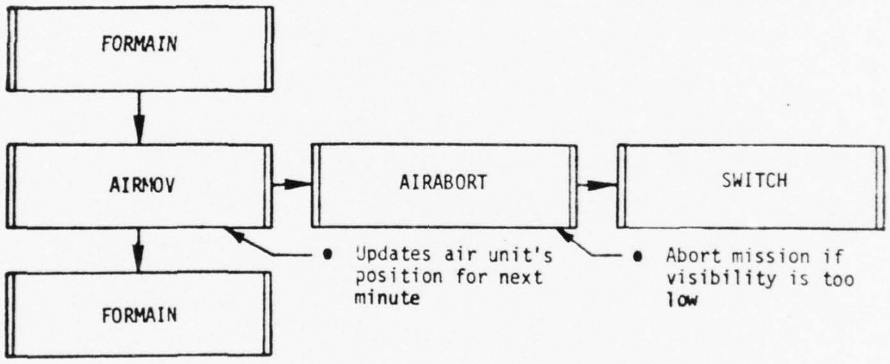

FORMAIN
AIRMOV
AIRABORT
SWITCH
FORMAIN
• Updates air unit's position for next minute
• Abort mission if visibility is too low

Figure 5-110. Air Module Subroutine Linkage for AIRMOV Chain

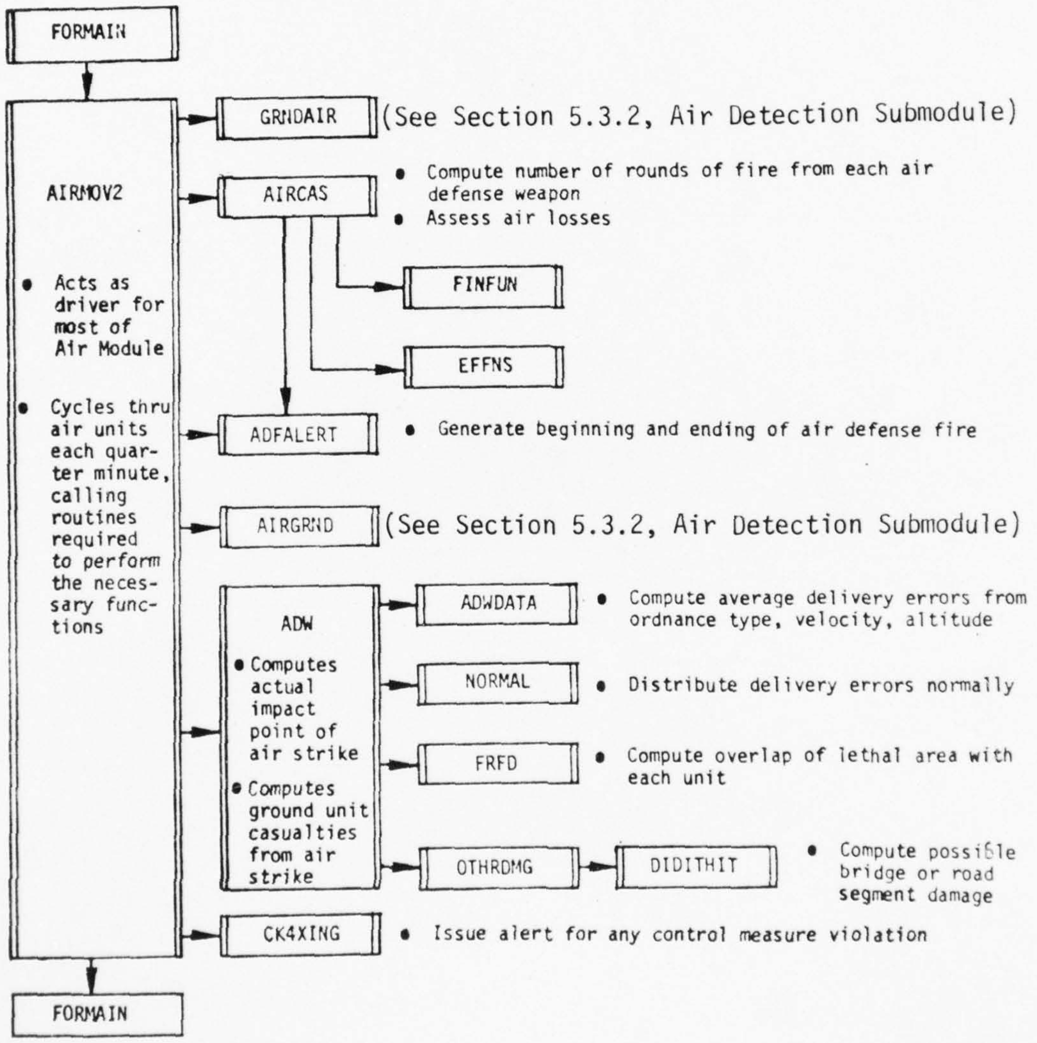

FORMAIN
AIRMOV2
• Acts as driver for most of Air Module
• Cycles thru air units each quarter minute, calling routines required to perform the necessary functions
GRNDAIR
(See Section 5.3.2, Air Detection Submodule)
AIRCAS
• Compute number of rounds of fire from each air defense weapon
• Assess air losses
FINFUN
EFFNS
ADFALERT
• Generate beginning and ending of air defense fire
AIRGRND
(See Section 5.3.2, Air Detection Submodule)
ADW
• Computes actual impact point of air strike
• Computes ground unit casualties from air strike
ADWDATA
• Compute average delivery errors from ordnance type, velocity, altitude
NORMAL
• Distribute delivery errors normally
FRFD
• Compute overlap of lethal area with each unit
OTHRDMG
DIDITHIT
• Compute possible bridge or road segment damage
CK4XING
• Issue alert for any control measure violation
FORMAIN

Figure 5-111. Air Module Subroutine Linkage for AIRMOV2 Chain

The Air Module begins with a call to AIREVENT by PPEVENT, the primary EVENT activator, when an air event notice has been created by a controller. The remainder of the Air Module chain begins with a call to AIRMOV2 by FORMAIN, the main mathematical model driver program. AIRMOV2 acts as a driver for the majority of the module by cycling through all air units and ground units each quarter minute, using the air unit positions pre-calculated by AIRMOV. It calls subroutines to determine ground-air and air-ground detections, air defense fire from ground units against air units and its effect, accomplishment of an air strike against a ground target and its effect, checks for control measure violations, and generates appropriate alert messages.

When an air unit is originally created and processed by AIREVENT, it generally is given a non-zero take-off delay (IAMTE(IAC,1)). When the take-off delay expires, AIRMOV2 will still not process the air unit because it has no position points until AIRMOV executes. Subsequently, AIRMOV is called by FORMAIN near the end of a time-step. It computes the air unit's position for the entire next time-step, at quarter minute intervals or less. When the next time-step occurs, AIRMOV2 will process pre-computed points. The cycle is repeated until the air unit completes its mission, at which time a landing delay is applied by AIRMOV and the air unit is henceforth ignored by the Air Module, except that it is kept active for display during the landing period. At the completion of the landing period, the air unit ceases to exist and is removed from active status by AIRMOV.

The subroutines are described individually with accompanying flow charts in the material that follows. The individual write-ups are grouped according to the linkage chain they belong to and their sequence in that chain.

5.8.1.1 AIREVENT Chain

AIREVENT is called by PPEVENT whenever a new air event notice is to be processed during the current time-step. The purpose of AIREVENT is to process the event notice so that the proper data for the particular mission desired has actually been included by the controller in the event notice. (See page 5-950 for an example of an air event notice (event type 8)).

Once that determination has been made, and also the determination that there are no conflicting requirements contained within the event notice, all the variables required by AIRMOV and AIRMOV2 are initialized to begin the air mission. AIREVENT does essentially the same checking whether the event notice is for a new mission or a modification of the old mission. If the event notice is a cancellation notice, then AIREVENT sets the appropriate variables, generates the alert, and calls AIRABORT without conducting any checking.

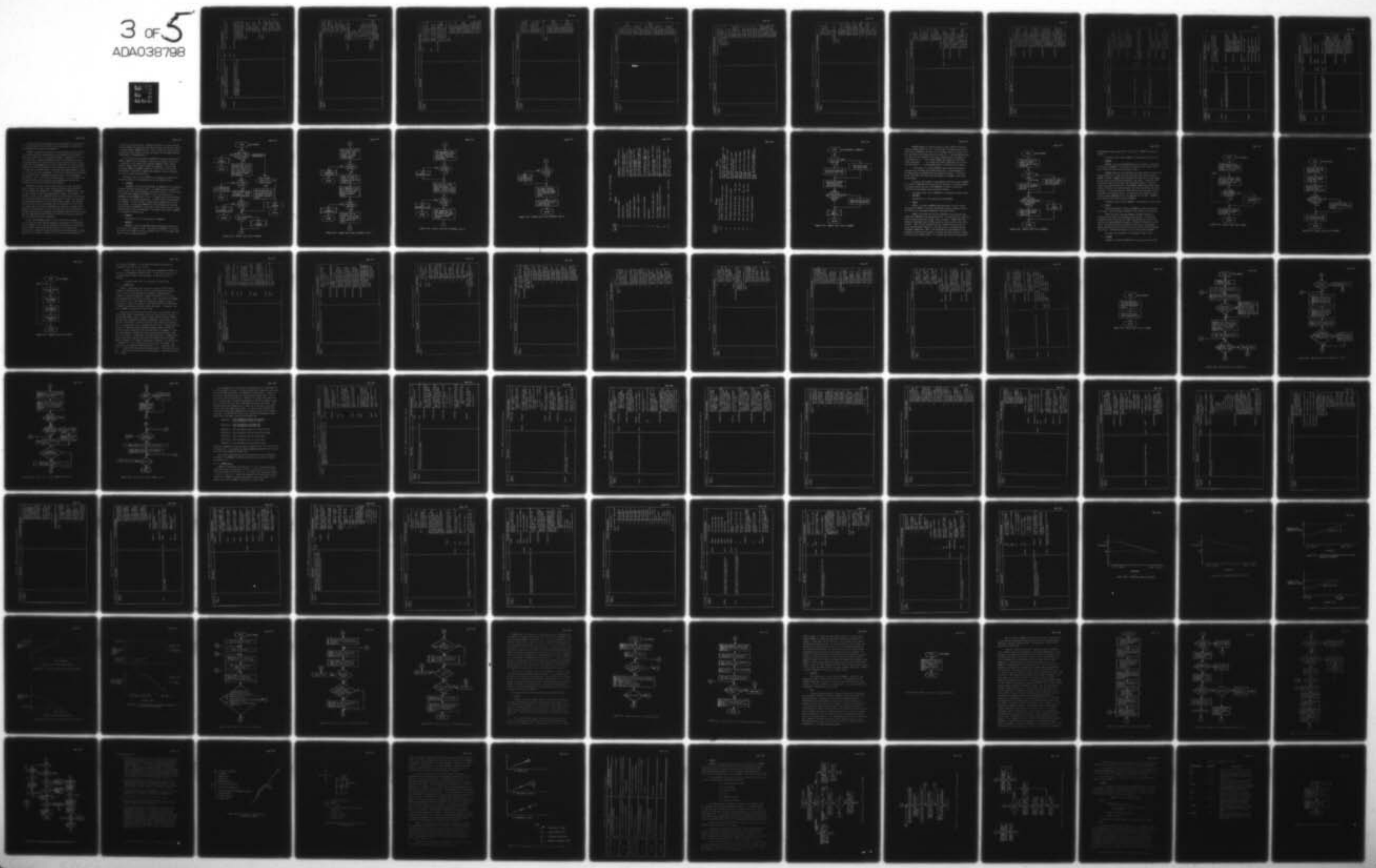
The AIREVENT chain subroutine linkage is shown in Figure 5-109. Those subroutines described in detail in other sections are so noted. Table 5-55 presents a brief description of each subroutine in AIREVENT, including the principal inputs and outputs. See Table 5-56 for a listing of the error messages by error type.

AIREVENT begins by processing the second word of the event notice (INVDT(2)) to determine the general type of the event notice (i.e., new mission, modify old mission or cancel mission). If the event notice is for the creation of a new air mission, AIREVENT next analyzes INVDT(3) to determine the nature of the mission (red or blue, strike or reconnaissance), checks to see if there are any air units available, and examines INVDT(4) and INVDT(5) to insure a proper equipment type and number of equipments have been specified. Using aircraft type, AIREVENT then determines if the meteorological visibility (VISM) and density altitude (DNSALT) are proper to allow continuance of the mission. If any of these tests are failed, AIREVENT calls AIRERROR to generate a RAMALERT for the instructor and returns to the calling routine. If none of the tests are failed, personnel are assigned to the air unit and AIREVENT begins its next portion of its analysis of the event notice.

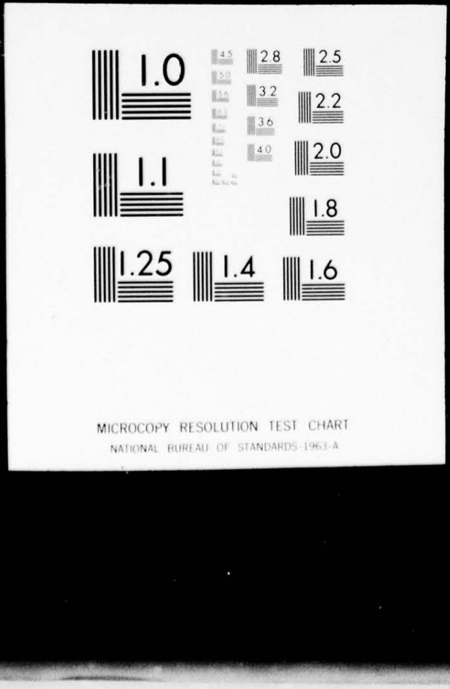
If the event notice was to modify an old mission, AIREVENT does some initial checking to determine the status of the old mission. If the old mission is still active, internal variables in AIREVENT are set from data contained in the event notice and the unit files for that mission, and the code branches to the next portion of the event notice analysis. If the old mission is no longer active, the mission is cancelled with a call to ALERTGEN to generate the RAMALERT, and AIREVENT returns to the calling notice.

AD-A038 798 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

3 OF 5
ADA038798



3 OF 5
ADA038798



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Table 5-55. AIREVENT Chain Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
PPEVENT	Activates the appropriate events processor in accordance with the supplied event type.	Input: INVDT(JEVT) Event notice data of event that is being processed. Output: None
AIREVENT	Analyzes air event notice to ensure the mission requirements do not conflict with aircraft performance capabilities, and initializes variables for AIRMOV and AIRMOV2.	Input: IAMTE(IEQ,IMODE) - For ground equipment (indicated by IEQCOD(IEQ) >0), ammunition type used by weapon type IEQ in mode of operation IMODE (0 implies no ammunition used). For aircraft (IEQCOD(IAC) = -1), IMODE = 1 Takeoff delay time for this aircraft (in minutes). = 2 Landing delay time for this aircraft (in minutes). For air ordnance (IEQCOD(IAC) = -2), IMODE = 1 Number of Ammunition type for this equipment, if any. = 2 Number of rounds in a standard load of that ammunition type. = 3 Rate of rie of this equipment (in rounds/ minute).

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		= 4 Number of drops per pass (applies to bombs). = 5 Distance between drops (in meters) = 6 Dud probability times 100 = 7 Kill probability times 100 for bridge type = 8 Kill probability times 100 for bridge type. ($1 \le IEQ \le 80$, $1 \le IMODE \le 8$) IEQCOD(IEQ) Equipment category code of given equipment type IEQ ($1 \le IEQ \le 80$): -3 Air sensor -2 Air weapon -1 Aircraft 0 Not a weapon 1 Direct fire weapon 2 Indirect fire weapon 3 Support fire weapon 4 Air defense weapon INVDT(JEVT) (See subroutine PPEVENT). IPVCE(IEQ,IMODE) - Personnel vulnerability class associated with each mode IMODE of equipment type IEQ. ($1 \le IEQ \le 80$, $1 \le IMODE \le 8$)

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		NFDU — Number of first dummy unit. (Dummy units include air units used to index air unit arrays.) Air unit index for unit I is I+1-NFDU. NLDU — Number of last dummy unit. (Dummy units include air units.) NLDU minus NFDU must be less than 10. ROFE(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ) $\geq$0), rate of fire of weapon type IEQ in each mode IMODE (in rounds/minute). For aircraft (IEQCOD(IEQ) = -1), IMODE = 1 Fuel expenditure for losing altitude (in pounds/meter). = 2 Fuel expenditure for gaining altitude (in pounds/meter). = 3 Fuel expenditure at minimum speed, minimum load, best pressure density (in pounds/minute). = 4 Fuel expenditure at cruise speed (in pounds/minute). = 5 Fuel expenditure at maximum speed (in pounds/minute).

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		= 6 Ratio of fuel expenditure rate at maximum load to fuel expenditure rate at minimum load. = 7 Ratio of fuel expenditure at worst pressure density to fuel expenditure rate at best pressure density. = 8 Not used For air ordnance (IEQCOD(IEQ) = -2), IMODE = 1 Fraction of personnel in personnel vulnerability class 1 (standing) and within target area who are killed by this equipment. = 2 Fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equipment.

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		= 3 Fraction of personnel in personnel vulnerability class 3 (prone) and within target area who are killed by this equipment. = 4 Fraction of equipment with IEQCLS=1 and within target area which is damaged by this equipment. = 5 Fraction of equipment with IEQCLS=2 and within target area which is damaged by this equipment. = 6 Fraction of equipment with IEQCLS=3 and within target area which is damaged by this equipment. = 7 Fraction of equipment with IEQCLS=4 and within target area which is damaged by this equipment. $1 \leq IEQ \leq 80$, $1 \leq IMODE \leq 8$)

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		ROME(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ) $\geq$0), rate of movement (in meters/minute) of weapon type I in mode IMODE (unobstructed). For aircraft (IEQCOD(IEQ)=-1), IMODE = 1 Maximum load aircraft can carry (in pounds) at best modeled pressure density (PDBEST). = 2 Maximum altitude of aircraft (in meters). = 3 Minimum speed of aircraft (in meters/minute). = 4 Cruise speed of aircraft (in meters/minute). = 5 Maximum speed of aircraft (in meters/minute). = 6 Maximum load aircraft can carry (in pounds) at worst modeled pressure density (PDWORST); the correct negative value will insure that an aircraft cannot fly at pressure densities below its capability.

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		= 7 Poorest meteorological visibility in which aircraft can continue its mission (in meters). = 8 Not used. For equipment other than aircraft (IEQCOD(IEQ) =-2 or -3), IMODE = 1 Weight of equipment (in pounds) including standard ammunition load. = 2 Minimum speed (in meters/minute) at which equipment can be used. = 3 Maximum speed (in meters/minute) at which equipment can be used. = 4 Minimum altitude at which equipment can be used (in meters). = 5 Maximum altitude at which equipment can be used (in meters). = 6 For sensors, not used; for ordnance, road

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		crater radius (in meters) against road type 1. = 7 For sensors, not used; for ordnance, road crater radius (in meters) against road type 2. = 8 For sensors, not used; for ordnance, road crater radius (in meters) against road type 3. ($1 \leq IEQ \leq 80$, $1 \leq MODE \leq 8$) Output: IACPV(JCKPT,JAU) - Velocity (in meters/minute) at JCKPT-th check point for air unit JAU. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IACPX(JCKPT,JAU) - X-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IACPY(JCKPT,JAU) - Y-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$)

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		IACPZ(JCKPT,JAU) - Z-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) INOG(JAU) Operational grouping to which unit JAU belongs. (If 0, unit JAU does not belong to any op group). IXAIR(JAIRTE,JAU) - X-ccordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPX). ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$) IYAIR(JAIRTE,JAU) - Y-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPY). ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$) IZAIR(JAIRTE,JAU) - Z-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPZ). ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$)

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description		Principal Inputs/Outputs	
AIREVENT (Cont'd)			NEXTACP(JAU)	Index within /AIROUTES/ arrays of next check point for air unit JAU. ($1 \le JAU \le 10$)
			TOTEQU(JAU,IEQ)	- Total number of pieces remaining of IEQ-th equipment type carried by unit JAU. ($1 \le JAU \le 100$, $1 \le IEQ \le 14$)
			USEEQU(JU,J)	Number of pieces manned for the J-th equipment type carried by unit JU. ($1 \le JU \le 100$, $1 \le J \le 14$)
AIRERROR	Formats error message for a RAMALERT.	Input:	None	
		Output:	Sends RAMALERT when called by Air Module after error in event created by menu has been found.	
RAMALERT	(See Section 5.7.3).			
BRRDCHCK	Determines target location as middle of road or bridge if target is road or bridge.	Input:	IBRIDGEX(IBR,M)	- First (M=1) and second (M=2) X coordinate of bridge IBR. ($1 \le IBR \le 16$)
			IBRIDGEY(IBR,M)	- First (M=1) and second (M=2) Y coordinate of bridge IBR. ($1 \le IBR \le 16$)
			INVDT(JEVT)	(See subroutine PPEVENT)
			IROADX(IRDSG)	The X coordinate of all the input roads. Index into array for road N given by

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs		
BRRDCHCK (Cont'd)				IRDSTRT(N). ($1 \leq IRDSG \leq 500$)
		Output:	INVDT(JEVT)	(See subroutine PPEVENT)
			(Calls AIRERROR to issue RAMALERT if no road or bridge is found within 1,000 meters of X,Y point.)	
ALERTGEN	(See Section 5.7.2.)			
ORDPRI	Searches through ordnance carried by aircraft and picks highest priority ordnance for target.	Input:	NPTEC(IEQ,J)	Equipment number of three primary target types for weapon *IEQ*; -1 implies a personnel target. ($1 \leq IEQ \leq 80$, $1 \leq J \leq 3$)
			NSTE(IEQ,J)	Equipment numbers of three secondary target types for weapon IEQ; -1 implies a personnel target. ($1 \leq IEQ \leq 80$, $1 \leq J \leq 3$)
			USEEQU(JU,J)	(See subroutine AIREVENT).
		Output:	IBETA	Weapon class number.
AIRABORT	Aborts air mission if visibility is too low.	Input:	IACPX(JCKPT,JAU) - (See subroutine AIREVENT.)	
			IACPY(JCKPT,JAU) - (See subroutine AIREVENT.)	
			IACPZ(JCKPT,JAU) - (See subroutine AIREVENT.)	
			NEXTACP(JAU)	(See subroutine AIREVENT)

Table 5-55. AIREVENT Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs		
AIRABORT (Cont'd)		Output:	AIRBORT(JAU)	Logical variable to indicate if an air unit mission has been aborted = .TRUE. if aborted; = .FALSE. if not. ($1 \leq JAU \leq 10$)
			NEXTACP(JAU)	(See Subroutine AIREVENT)
SWITCH	Utility routine to reverse entries in an array.	Input:	IDUMMY	Dummy argument
		Output:	IDUMMY	Dummy argument
CLEARDET	Clears ground-to-air and air to ground detection *arrays, and clears ground-to-air engagement* array.	Input:	IAU1	Number of air unit.
		Output:	IAGDET(JU,JAU,JAS)	- Bit maxtrix indicating whether ground unit IGU has already been detected by air unit JAU with sensor JAS. IGU is used to determine JU. ($1 \leq JU \leq 4$, $1 \leq JAU \leq 10$, $1 \leq JAS \leq 6$)
			IGADET(JU,JAU)	- Bit matrix indicating whether ground unit IGU has already detected air unit JAU. IGU is used to determine JU. ($1 \leq JU \leq 4$, $1 \leq JAU \leq 10$)
			IPRVENG(JU,JAU)	- Bit matrix indicating whether ground unit IGU is currently engaging air unit JAU with air defense weapons IGU is used to determine JU. ($1 \leq JU \leq 4$, $1 \leq JAU \leq 10$)

At this point, the two threads of code associated with a new mission or the modification of an old mission have come together in AIREVENT for the remainder of the event notice analysis.

AIREVENT now begins the processing of the selected route to determine any incorrect entries, inconsistencies or requirements for performance beyond the capabilities of the aircraft selected. AIREVENT calls, if this is a new mission, subroutine CLEARDET to clear all the detection arrays indicating detection of ground unit by air unit, detection of air unit by ground unit, and engagement of air unit by ground unit air defense weapons. AIREVENT then checks the number of points selected on the route of flight to insure they are within limits, it analyzes the target selected to insure it is valid, and finally determines if visibility conditions are appropriate for a strike mission. AIREVENT also checks to see if a guided weapon (IEQCLS(IEQ) = 10) is part of the aircraft weapons. If so, it assumes that the target selected is the target of the guided weapon, and does the remainder of its processing based on that premise.

AIREVENT next begins a point by point analysis of the route during which it checks each leg to insure that altitude and speed limitations are not exceeded (IV < ROME(IAC,3), IV > ROME(IAC,5), IZ > ROME(IAC,2)). When it checks the leg of the route containing the target, if the target is a bridge or a road, it calls BRRDCHCK to get the coordinates of the mid-point of the road or bridge to use as the target point. It checks to insure that fuel expenditure rates are not less than minimum assigned. Then AIREVENT conducts an analysis of the selected load to insure that it is less than the maximum that can be carried (MAXLOAD) and that there is enough fuel on board to fly the new route (if this is a modification of an old mission). AIREVENT then examines the equipment load (ordnance and/or sensors) to insure they are compatible with the aircraft, and that the aircraft can actually carry the total weight of the equipment.

At this point AIREVENT checks to insure that there is ordnance if it is a strike mission, and that the ordnance is appropriate for the target. It calls ORDPRI to pick, from the ordnance on the aircraft, the particular kind of ordnance that has the highest priority for the target type. If the mission is a stand-off strike, but there are no guided weapons on board, there will be a ram-alert generated for the instructor and AIREVENT will return to the calling routine. AIREVENT then checks to insure that the distance between the stand-off point and the target is within the range of the guided weapon (IMAXRE(IBETA,1)). There are then four checks against minimum and maximum release altitudes and speeds for the weapon selected (ROME(IBETA,2) to ROME(IBETA,5)).

If all the tests are passed then AIREVENT initializes variables for AIRMOV, AIRMOV2 and the foreground. Most of the variables initialized are contained in the common blocks ROUTING, MOVERATE, BSTAT, AIRLOC, AIROUTES, BOBBY, BLEFT1, BLEFT2, AVFUEL and FGAIRLOC. They are used throughout AIRMOV and AIRMOV2 during the "flight" of the air mission. The variables initialized in common block FGAIRLOC are used to display the route of flight on the monitors.

The general logic flow of AIREVENT is illustrated in Figure 5-112. Brief descriptions of the subroutines called by AIREVENT follow.

a. AIRERROR

AIRERROR is called by both AIREVENT and BRRDCHCK with a subroutine argument indicating the type of error. It formats the error message for the particular error type detected by the calling routines. It first gets the mission name (INVDT(6) and INVDT(7) if this is a new mission, IUNNAME(1,IU) and IUNNAME(2,IU) if this is a modification or cancellation) and combines it with the error message. If there is an invalid equipment type (IERROR = 5 or 6) involved, it merges the equipment type into the error message. It then calls RAMALERT to send the message to the appropriate instructor (INVDT(64)), and returns to the calling routine. Table 5-56 shows the various error messages. The general logic flow of AIRERROR is illustrated in Figure 5-113.

b. RAMALERT

See Section 5.7.3 for a description of RAMALERT.

c. BRRDCHCK

BRRDCHCK is called by AIREVENT when the designated target of an air mission is a bridge or a road. The purpose of BRRDCHCK is to return the center of a bridge or road as the target point if it lies within 1000 meters of the selected target point.

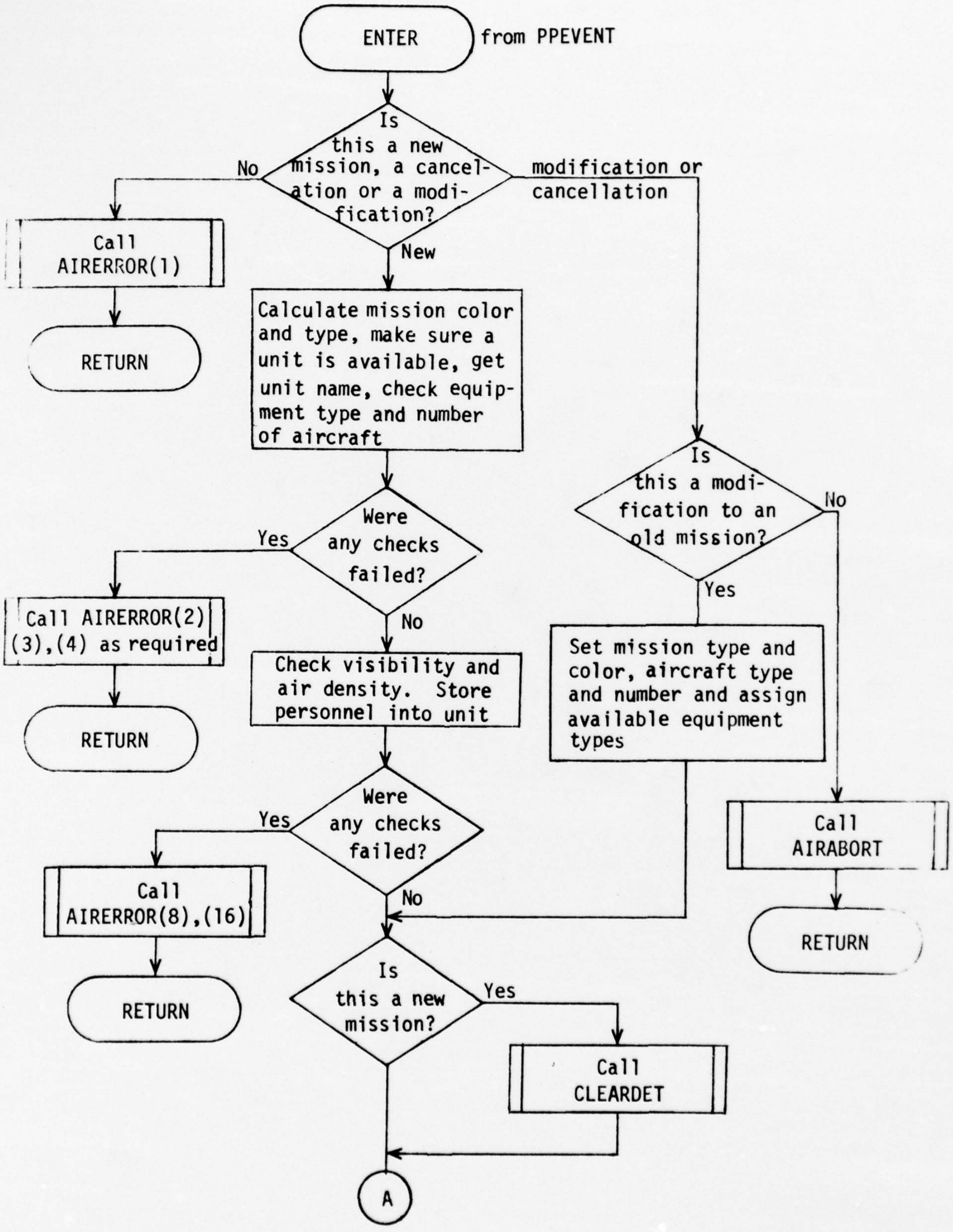

ENTER
from PPEVENT
Is this a new mission, a cancellation or a modification?
No
modification or cancellation
New
Call AIRERROR(1)
RETURN
Calculate mission color and type, make sure a unit is available, get unit name, check equipment type and number of aircraft
Is this a modification to an old mission?
No
Were any checks failed?
Yes
Call AIRERROR(2) (3),(4) as required
RETURN
No
Check visibility and air density. Store personnel into unit
Yes
Set mission type and color, aircraft type and number and assign available equipment types
Were any checks failed?
Yes
Call AIRERROR(8),(16)
RETURN
No
Call AIRABORT
RETURN
Is this a new mission?
Yes
Call CLEARDET
A

Figure 5-112. General Logic Flow of AIREVENT

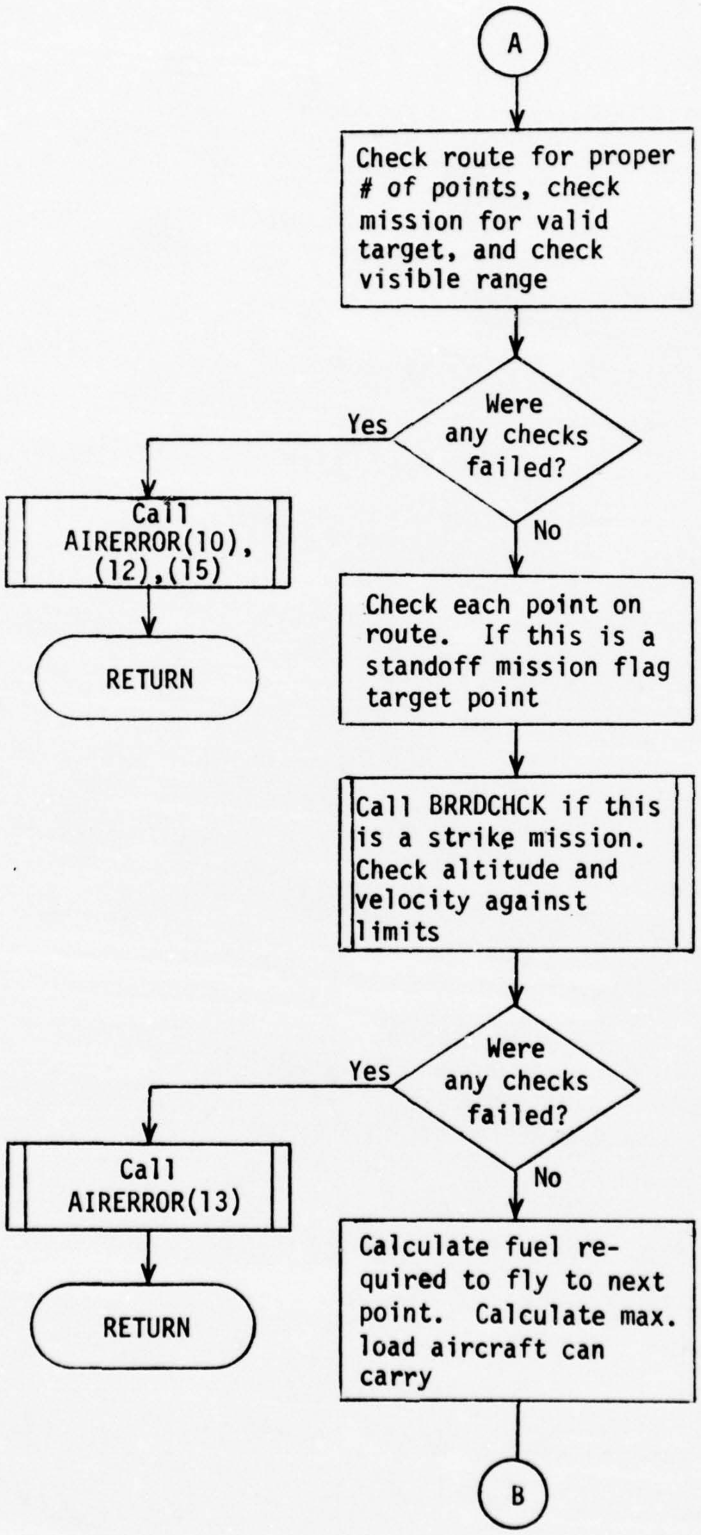

A
Check route for proper # of points, check mission for valid target, and check visible range
Were any checks failed?
Yes
No
Call AIRERROR(10), (12),(15)
RETURN
Check each point on route. If this is a standoff mission flag target point
Call BRRDCHCK if this is a strike mission. Check altitude and velocity against limits
Were any checks failed?
Yes
No
Call AIRERROR(13)
RETURN
Calculate fuel required to fly to next point. Calculate max. load aircraft can carry
B

Figure 5-112. General Logic Flow of AIREVENT, Cont'd.

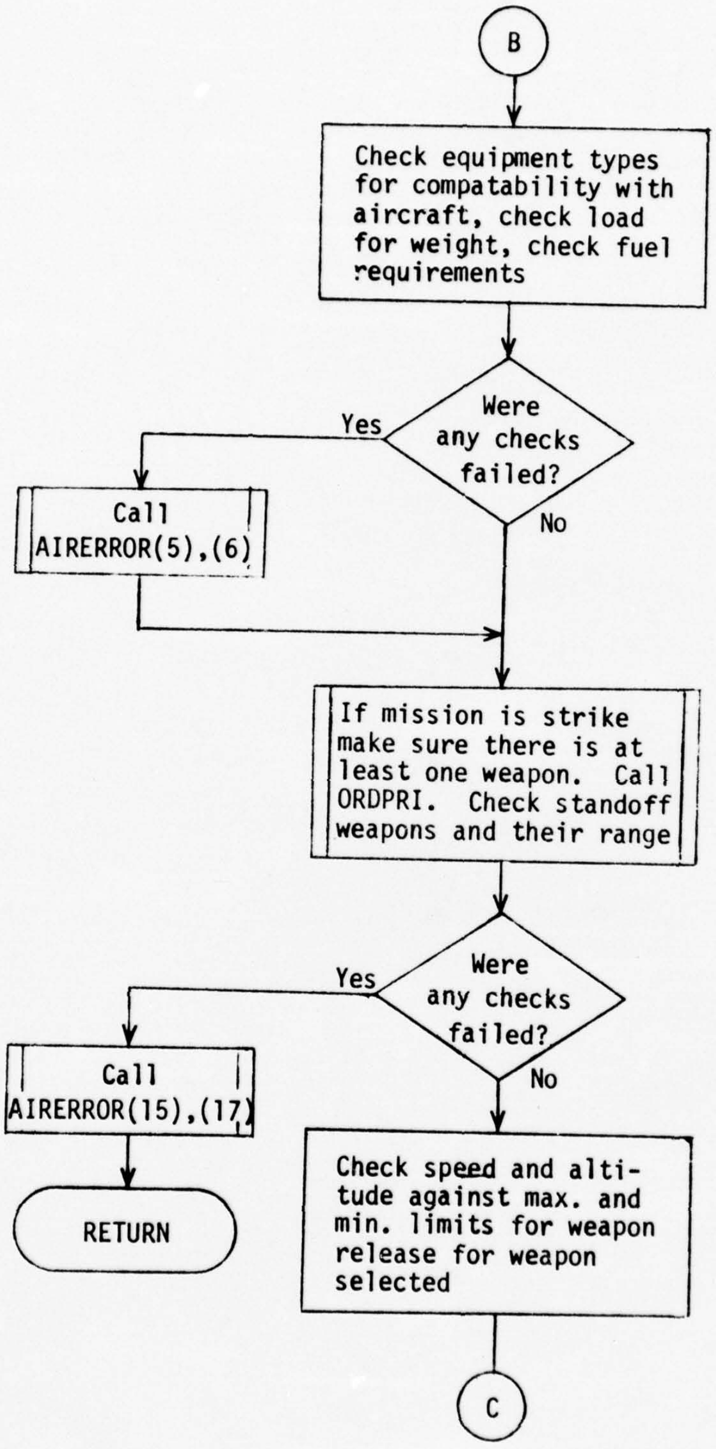

B
Check equipment types for compatability with aircraft, check load for weight, check fuel requirements
Were any checks failed?
Yes
No
Call AIRERROR(5),(6)
If mission is strike make sure there is at least one weapon. Call ORDPRI. Check standoff weapons and their range
Were any checks failed?
Yes
No
Call AIRERROR(15),(17)
RETURN
Check speed and altitude against max. and min. limits for weapon release for weapon selected
C

Figure 5-112. General Logic Flow of AIREVENT, Cont'd

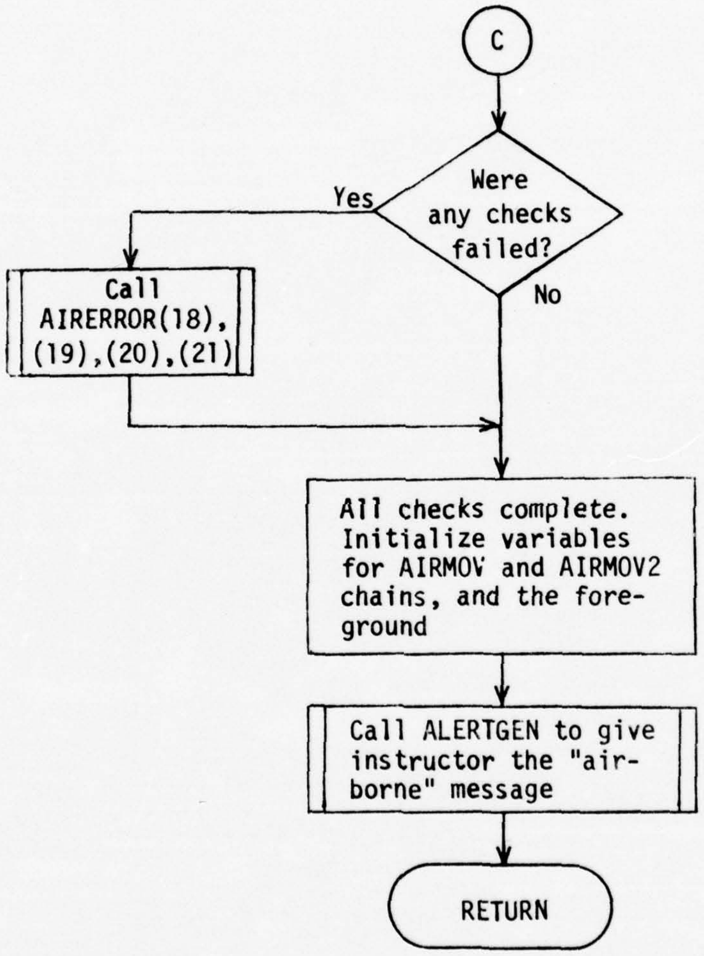

C
Were any checks failed?
Yes
No
Call AIRERROR(18), (19),(20),(21)
All checks complete. Initialize variables for AIRMOV and AIRMOV2 chains, and the fore- ground
Call ALERTGEN to give instructor the "air- borne" message
RETURN

Figure 5-112. General Logic Flow of AIREVENT, Cont'd

Table 5-56. Air Error Messages

Air Error Type (#)	Message	Problem
1	Invalid event subtype	Second word of event notice not a 1, 2, or 3.
2	Invalid mission	Third word of event notice not a 1, 2, 5 or 6 or invalid target.
3	No available units	All dummy air units are in use.
4	Invalid aircraft type	Aircraft selected is not in equipment deck, or no A/C in flight.
5	Invalid equipment deleted - -	Not an A/C equipment or A/C can't carry this equipment.
6	Load exceeded - - Deleted EQ	Total load exceeds A/C capability or fuel requirement exceeds capacity.
7	Invalid old unit selected	Unit selected not a dummy air unit or air unit not in proper status.
8	Bad weather	Visibility below minimum for A/C type.
9	Fuel required exceeds capacity	Fuel load too large or fuel required exceeds current load.
10	Invalid number of route points	Number of points on route <2 or >9.
11	Invalid ordnance for standoff delivery	Weapon selected not guided (IEQCLS ≠ 10).
12	Invalid target	Improper target type or target point is landing point.
13	Aircraft performance exceeded - - Limit used	Altitude is >A/C limit, or speed is > max. or < min. limits.

Table 5-56. Air Error Messages (Cont'd)

Air Error Type (#)	Message	Problem
14	Mission already cancelled	Variable airbort = true
15	Standoff air delivery point out of range	Visibility $<$eye visual range or slant range too long or no weapons.
16	Density altitude above aircraft capability	Calculated density altitude $>$input A/C limit.
17	No ordnance available for this target	No weapon in flight can be used against selected target.
18	Min. speed too low for equip. - - Min. used	Speed too low to release weapon selected.
19	Max. speed too high for equip. - - Max. used	Speed too high to release weapon selected.
20	Min. alt. too low for equip. - - Min. used	Altitude too low to release weapon selected.
21	Max. alt. too high for equip. - - Max. used	Altitude too high to release weapon selected.
22	No target at designated point	Center of nearest road segment or bridge too far from designated target point.

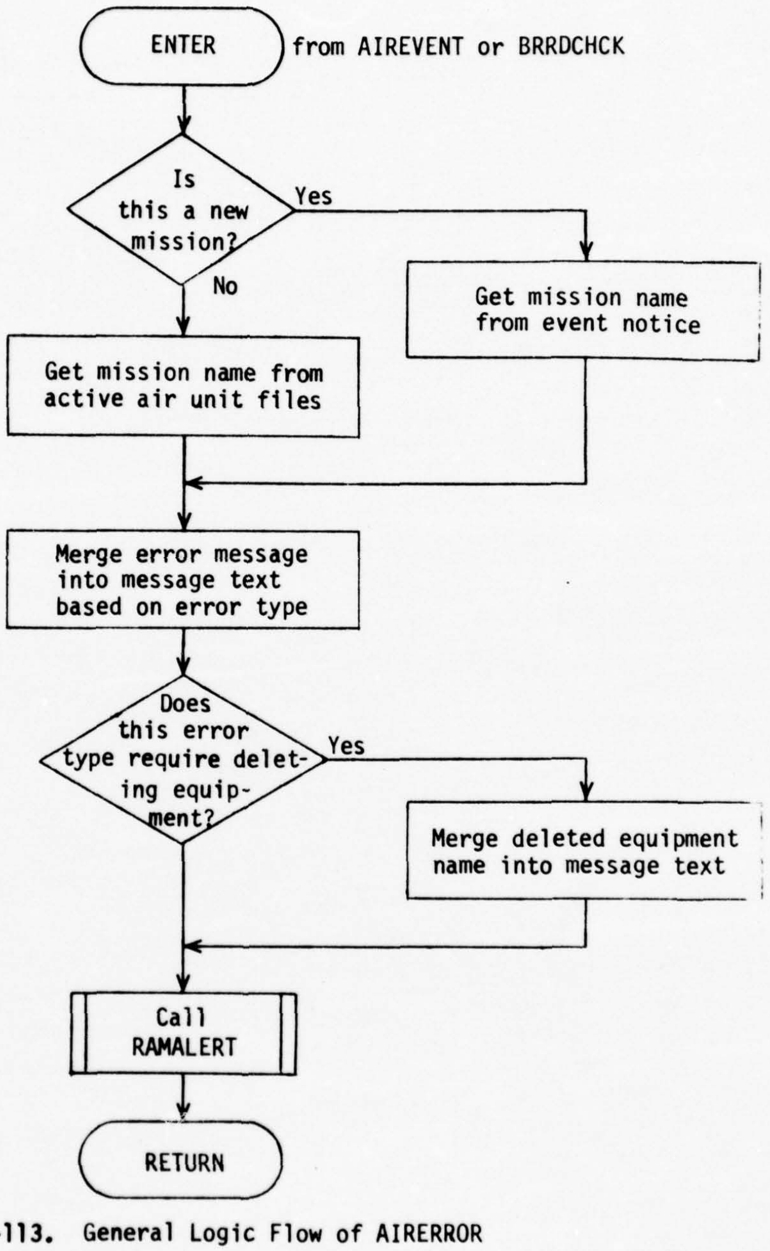

ENTER
from AIREVENT or BRRDCHCK
Is this a new mission?
Yes
No
Get mission name from event notice
Get mission name from active air unit files
Merge error message into message text based on error type
Does this error type require deleting equipment?
Yes
Merge deleted equipment name into message text
Call RAMALERT
RETURN

Figure 5-113. General Logic Flow of AIRERROR

BRRDCHCK begins by initializing the x- and y-values of the test distance (ISAVEX,ISAVEY) to 1001. It then checks the target type from the event notice (INVDT(29)) to insure it is a bridge, road or x-y point, and gets the index to the target point for future use. If the designated target is a road, it loops through all the roads, checking each road segment by calculating the center point of each segment and comparing it with the target coordinates. The last center point whose coordinates are both less than 1001 meters from the designated target is then used as the target.

If the target is a bridge BRRDCHCK loops through the bridges in the model and calculates each bridge center point. The coordinates of the center point are compared with the target coordinates. If both are within 1001 meters then the target coordinates are changed to the center of the bridge.

As long as some bridge or road segment satisfies the test, BRRDCHCK will return to AIREVENT and the mission will continue. If no road segment or bridge satisfies the test, then AIRERROR is called to inform the instructor of the problem, and the mission continues.

The general logic flow of BRRDCHCK is illustrated in Figure 5-114.

d. ALERTGEN

See Section 5.7.2 for a description of ALERTGEN.

e. ORDPRI

ORDPRI is called by AIREVENT when the mission is an air strike. The purpose of ORDPRI is to select the highest priority ordnance on board the aircraft for the target designated in the event notice.

ORDPRI initializes the priority (IFRSTPRI) to zero and gets the target type (INVDT(29)) from the event notice. It then sets up an index (K) based on target type (unit, bridge, road, x-y point). ORDPRI then loops through the equipment types (ITEQU) carried on the aircraft for the strike missions. When it finds a weapon that has a greater than zero priority against the designated target, it saves the priority and weapon type for subsequent comparisons. When it has completed the loop through all the equipments carried by the aircraft, it assigns the weapon type with highest priority to the variable IBETA. If no weapon qualifies to be used against

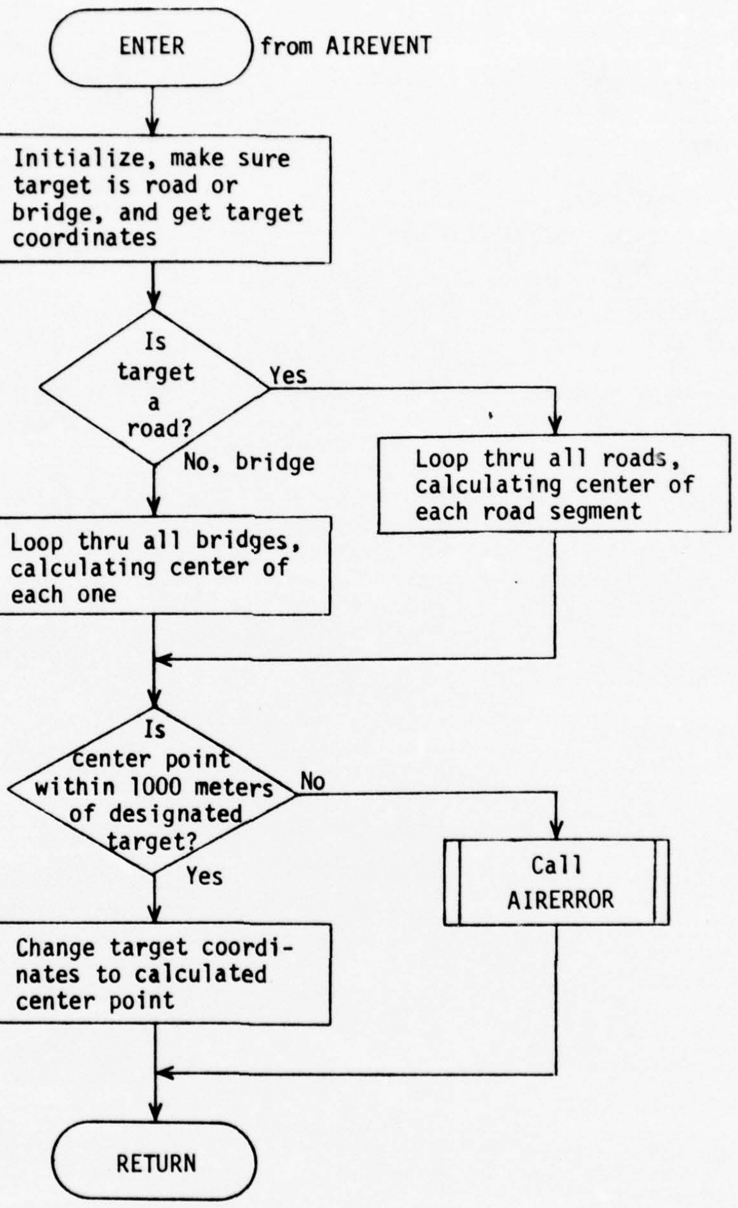

ENTER
from AIREVENT
Initialize, make sure target is road or bridge, and get target coordinates
Is target a road?
Yes
No, bridge
Loop thru all roads, calculating center of each road segment
Loop thru all bridges, calculating center of each one
Is center point within 1000 meters of designated target?
No
Yes
Call AIRERROR
Change target coordinates to calculated center point
RETURN

Figure 5-114. General Logic Flow of BRRDCHCK

the designated target, then IBETA is set to zero. ORDPRI then returns to AIREVENT.

The general logic flow of ORDPRI is illustrated in Figure 5-115.

f. AIRABORT

AIRABORT is called by AIREVENT when the air mission is cancelled. The purpose of AIRABORT is to set the air abort conditions flag to TRUE, and to select the route to be flown back to base.

AIRABORT begins by calculating the unit index (IAU) for the mission that was cancelled. It then sets the condition flag (logical variable AIRBORT) to TRUE. AIRABORT then calculates the square of the route length, and the square of the distance remaining on the route. If more than half of the route remains to be flown, then subroutine SWITCH is called to reverse the order of the points in the AIROUTES common block (IACPX,IACPY, IACPZ) so the aircraft actually reverses its course and retraces the route of flight. If less than half the route remains to be flown the aircraft continues on the selected route. In either case, the airspeed is set to cruise airspeed (ROME(IAC,4)) for the remainder of the route. AIRABORT then returns to the calling routine, AIREVENT.

The general logic flow of AIRABORT is illustrated in Figure 5-116.

g. SWITCH

Subroutine SWITCH is called by AIRABORT to reverse the entries in an array. SWITCH presumes a maximum of nine entries to be reversed.

SWITCH utilizes a one-word temporary storage location as a holding point for one of the variables being reversed. It begins by putting word #9 of the array into ITEMP, then moving word #1 to the #9 location, and then putting the contents of ITEMP into location #1 of the array. The switch of entries #1 and #9 of the array is now complete. SWITCH then does entries #2 and #8, #3 and #7, and #4 and #6 to complete the reversal. SWITCH then returns to the calling routine, AIRABORT.

The logic flow of SWITCH is illustrated in Figure 5-117.

h. CLEARDET

CLEARDET is called by AIREVENT once for every new air mission.

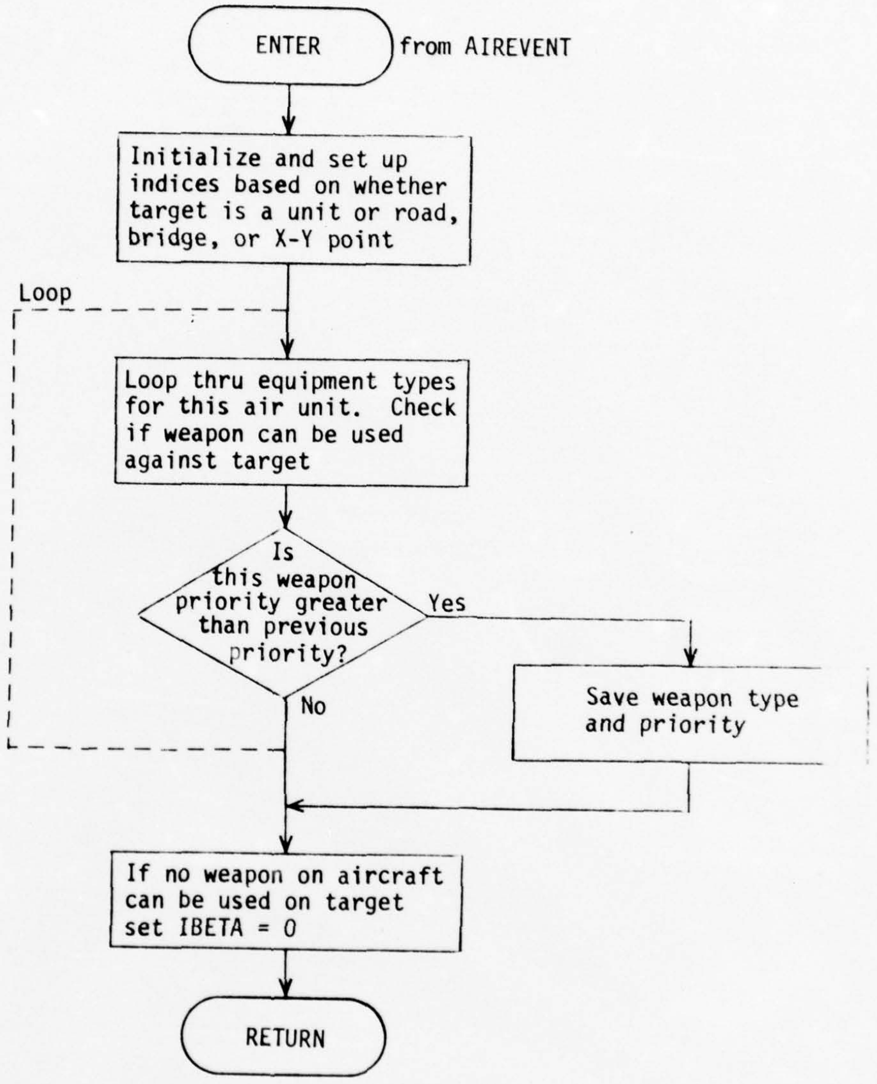

ENTER
from AIREVENT
Initialize and set up indices based on whether target is a unit or road, bridge, or X-Y point
Loop
Loop thru equipment types for this air unit. Check if weapon can be used against target
Is this weapon priority greater than previous priority?
Yes
No
Save weapon type and priority
If no weapon on aircraft can be used on target set IBETA = 0
RETURN

Figure 5-115. General Logic Flow of ORDPRI

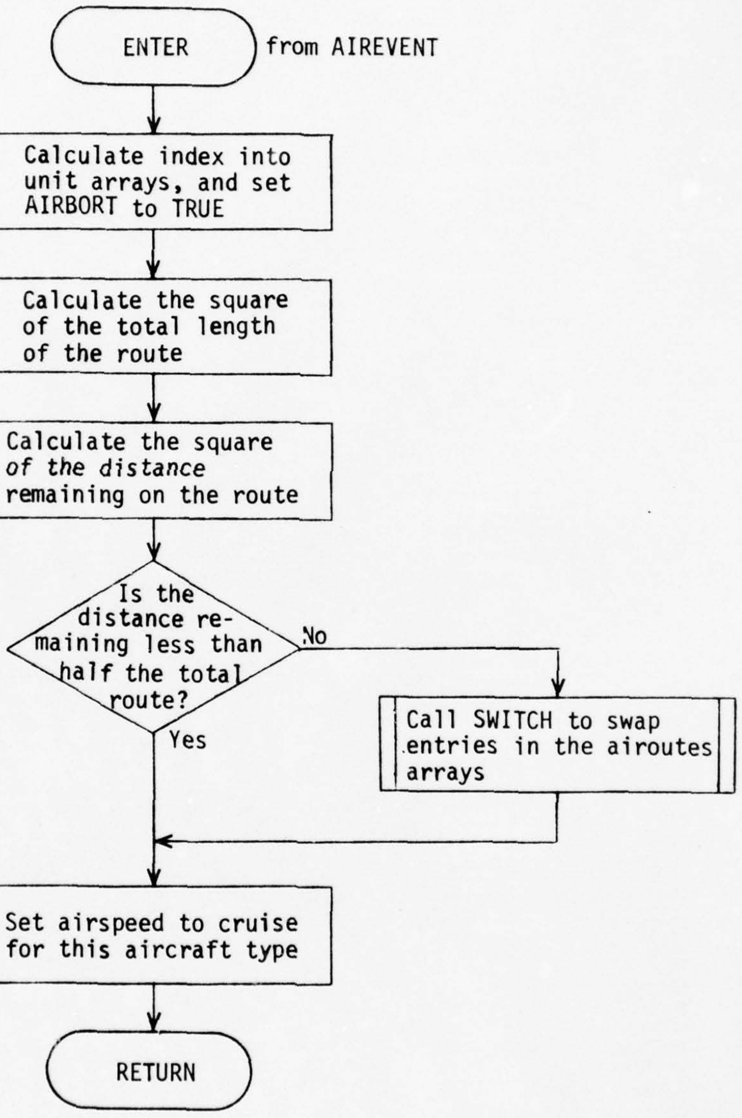

ENTER
from AIREVENT
Calculate index into unit arrays, and set AIRBORT to TRUE
Calculate the square of the total length of the route
Calculate the square of the distance remaining on the route
Is the distance remaining less than half the total route?
No
Yes
Call SWITCH to swap entries in the airoutes arrays
Set airspeed to cruise for this aircraft type
RETURN

Figure 5-116. General Logic Flow of AIRABORT

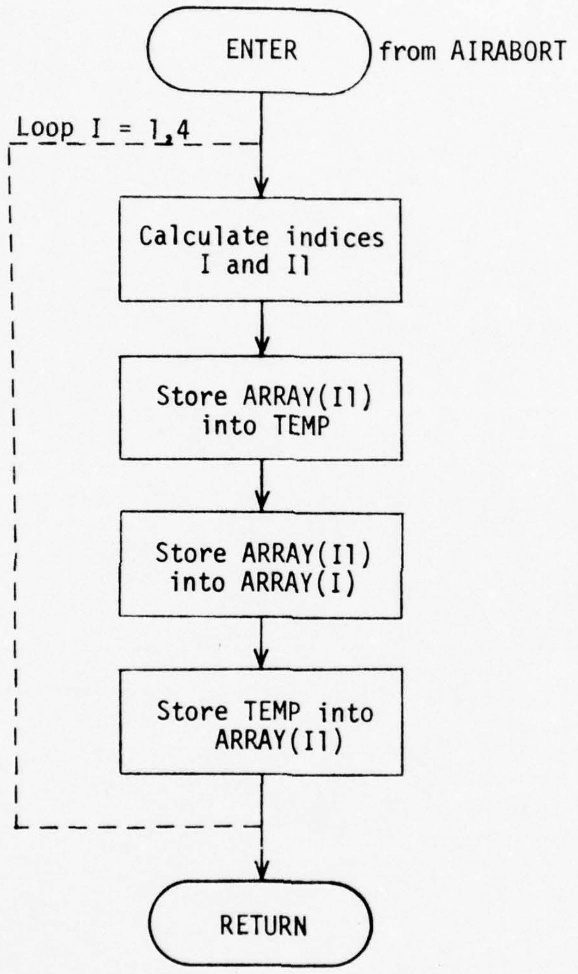

ENTER
from AIRABORT
Loop I = 1,4
Calculate indices I and I1
Store ARRAY(I1) into TEMP
Store ARRAY(I1) into ARRAY(I)
Store TEMP into ARRAY(I1)
RETURN

Figure 5-117. General Logic Flow of SWITCH

The purpose of CLEARDET is to initialize the detection and previous engagement arrays for this air mission.

CLEARDET sets to zero the ground-to-air engagements (IGADET) and the previous engagements (IPRVENG) arrays. It then sets to zero the air-to-ground sensor (IAGDET) array and returns to the calling routine, AIREVENT.

The general logic flow is illustrated in Figure 5-118.

5.8.1.2 AIRMOV Chain

AIRMOV is called once each time-step by the main driver program FORMAIN to compute positions for each air unit during the next minute. If the mission is to be aborted due to weather or command/control, AIRABORT is called to accomplish a change in direction if the air unit has completed less than half of its route. (For a description of AIRABORT, see Section 5.8.1.1) The AIRMOV chain subroutine linkage is shown in Figure 5-110. Table 5-57 presents a brief description of each subroutine in AIRMOV, including the principal inputs and outputs. The general logic flow of AIRMOV is presented in Figure 5-119.

Initially, the last position of the air unit in the previous time-step (IXAIR,IYAIR,IZAIR) is transferred to the first position this time-step (IXOLD,IYOLD,IZOLD). Next, visibility (VISM), density altitude (DNSALT), and fuel level (CLAVGAS) are checked to determine if the mission is to be aborted (visibility) or if the air unit will crash (density altitude and fuel). An appropriate alert is generated for each case. Assuming that the air unit is continuing, the differential between current altitude and velocity and that specified at the next check point is computed. If the next check point is a target unit which could be moving, the next check point is corrected. The distance to the next check point is compared with the distance to be covered to the end of the quarter minute at present speed to determine if a change in direction is required. If a check point is reached first, time of arrival at the check point is computed. The whole process of computing differentials for X, Y, Z to the next check point is repeated for the entire one minute interval. Fuel used, which had been computed for each subinterval and accumulated, is removed from the air unit's supply.

Table 5-57. AIRMOV Chain Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs		
FORMAIN	Secondary main program of the mathematical model. FORMAIN controls the minute-to-minute function flow between modules and submodules of the mathematical model.	Input:	None.	
		Output:	BDIR(JU,J)	Sine and cosine of angle faced by unit JU during preceding timestep (Cartesian coordinate system).
			BROMU(JU)	Rate of movement of unit JU (in meters/minute) for the preceding timestep.
			ITIME	Current time (initial time at input).
			ITIMED	JTIMED (in same common) minus ITIME. Used by subroutine FORSIG to calculate command and control event activation times (in minutes) since simulation start.
			JTIMED	Clock time (in minutes) corresponding to start of simulation.
			KTIME	Time step number.
			NDAYE	Calendar date of first day of simulation. The lowest numbered NDAYE for a set of exercises should be used as the first day of the set and used to set NDAYEMN1.
			NHOURE	Number of elapsed hours since midnight (0-23).
			NMINE	Number of elapsed minutes (0-59)
			TIME	Time (in floating point minutes).

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV	Calculates the new locations for air units.	Input: AIRBORT(JAU) Logical variable to indicate if an air unit mission has been aborted. ($1 \leq JAU \leq 10$). = TRUE if aborted = FALSE if not IACPV(JCKPT,JAU) Velocity (in meters/minutes) at JCKPT-th check point for air unit JAU. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IACPX(JCKPT,JAU) X-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IACPY(JCKPT,JAU) Y-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IACPZ(JCKPT,JAU) Z-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 10$) IAMTE(IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ)$\geq$0), ammunition type used by weapon type IEQ in mode of operation IMODE (0 implies no ammunition used). For aircraft (IEQCOD(IAC)=-1), IMODE = 1 Takeoff delay time for this aircraft (in minutes).

Table 5- 57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV		IMODE = 2 Landing delay time for this aircraft (in minutes). For air ordnance (IEQCOD(IAC) =-2), IMODE = 1 Number of ammunition type for this equipment, if any. = 2 Number of rounds in a standard load of that ammunition type. = 3 Rate of fire of this *equipment (in rounds/* minute). = 4 Number of drops per pass (applies to bombs). = 5 Distance between drops (in meters). = 6 Dud probability times 100. = 7 Kill probability times 100 for bridge type. = 8 Kill probability times 100 for bridge type. ($1 \leq IEQ \leq 80$, $1 \leq IMODE \leq 8$) NEXTACP(JAU) Index within /AIROUTES/ arrays of next check point for air unit JAU. ($1 \leq JAU \leq 10$)

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		NTGTPT(JAU) - Index within/AIROUTES/arrays of target point number for air unit JAU. ($1 \le JAU \le 10$) ROFE(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ)≥1), rate of fire of weapon type IEQ in each mode IMODE (in rounds/minute). For aircraft (IEQCOD(IEQ)=-1), IMODE = 1 Fuel expenditure for losing altitude (in pounds/meter). = 2 Fuel expenditure for gaining altitude (in pounds/meter). = 3 Fuel expenditure at minimum speed, minimum load, best pressure density (in pounds/minute). = 4 Fuel expenditure at cruise speed (in pounds/minute). = 5 Fuel expenditure at maximum speed (in pounds/minute). = 6 Ratio of fuel expenditure rate at maximum load to fuel expenditure rate at minimum load. = 7 Ratio of fuel expenditure rate at worst pressure density to fuel expenditure rate at best pressure density.

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		= 8 Not used. For air ordnance (IEQCOD(IEQ)=-2), IMODE = 1 Fraction of personnel in personnel vulnerability class 1 (standing) and within target area who are killed by this equipment. = 2 Fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equip- = 3 Fraction of personnel in personnel vulnerability class 3 (prone) and within target area who are killed by this equipment. = 4 Fraction of equipment with IEQCLS=1 and within target area which is damaged by this equipment. = 5 Fraction of equipment with IEQCLS=2 and within target area which is damaged by this equipment. = 6 Fraction of equipment with IEQCLS=3 and within target area which is damaged by this equipment.

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		= 7 Fraction of equipment with IEQCLS=4 and within target area which is damaged by this equipment. = 8 Fraction of equipment with IEQCLS=5 and within target area which is damaged by this equipment. (1≤IEQ≤80, 1≤IMODE≤8) ROME(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ)≥0), rate of movement (*in meters/minute*) of weapon type I in mode IMODE (unobstructed). For aircraft (IEQCOD(IEQ)=-1), IMODE = 1 Maximum load aircraft can carry (in pounds) at best modeled pressure density (PDBEST). = 2 Maximum altitude of aircraft (in meters). = 3 Minimum speed of aircraft (in meters/minute). = 4 Cruise speed of aircraft (in meters/minute). = 5 Maximum speed of aircraft (in meters/minute).

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		= 6 Maximum load aircraft can carry (in pounds) at worst modeled pressure density (PDWORST); the correct negative value will insure that an aircraft cannot fly at pressure densities below its capability. = 7 Poorest meteorological visibility in which aircraft can continue its mission (in meters). = 8 *Not used.* For equipment other than aircraft, (IEQCOD(IEQ)=-2 or -3), IMODE = 1 Weight of equipment (in pounds) including standard ammunition load. = 2 Minimum speed (in meters/minute) at which equipment can be used. = 3 Maximum speed (in meters/minute) at which equipment can be used. = 4 Minimum altitude at which equipment can be used (in meters). = 5 Maximum altitude at which equipment can be used (in meters).

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		= 6 For sensors, not used; for ordnance, road crater radius (in meters) against road type 1. = 7 For sensors, not used; for ordnance, road crater radius (in meters) against road type 2. = 8 For sensors, not used; for ordnance, road crater radius (in meters) against road type 3. (1≤IEQ≤80, 1≤IMODE≤8) VISM - Global meteorological visibility (in meters). Output: CLAVGAS(JU) - Current load of aviation fuel (in gallons) for unit JU. (1≤JU≤100) IXAIR(JAIRTE,JAU) - X-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPX). (1≤JAIRTE≤8, 1≤JAU≤10) IXY(IU,J) - Actual position coordinates (J=1 for X, J=2 for Y) of unit IU. (1≤IU≤100) IYAIR(JAIRTE,JAU) - Y-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute

Table 5-57. AIRMOV Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV (Cont'd)		plus 0 to 4 intermediate check points (IACPY). ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$) IZAIR(JAIRTE,JAU) - Z-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPZ). ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$) NEXTACP(JAU) - (See "Input" for this subroutine.) TIMXY(JAIRTE,JAU) - Time at which air unit JAU reaches JAIRTE-th point in this minute. ($1 \leq JAIRTE \leq 8$, $1 \leq JAU \leq 10$)

AIRABORT	Aborts air mission if visibility is too low.	Input: IACPX(JCKPT,JAU) - (See subroutine AIRMOV.)
		IACPY(JCKPT,JAU) - (See subroutine AIRMOV.)
		IACPZ(JCKPT,JAU) - (See subroutine AIRMOV.)
		NEXTACP(JAU) - (See subroutine AIRMOV.)
		Output: AIRBORT(JAU) - (See subroutine AIRMOV.)

SWITCH	Utility routine to reverse entries in an array.	Input: IDUMMY - Dummy argument. Output: IDUMMY - Dummy argument.

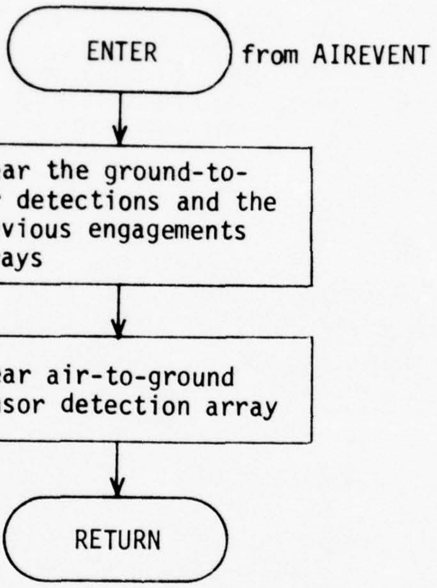

ENTER
from AIREVENT
Clear the ground-to-air detections and the previous engagements arrays
Clear air-to-ground sensor detection array
RETURN

Figure 5-118. General Logic Flow of CLEARDET

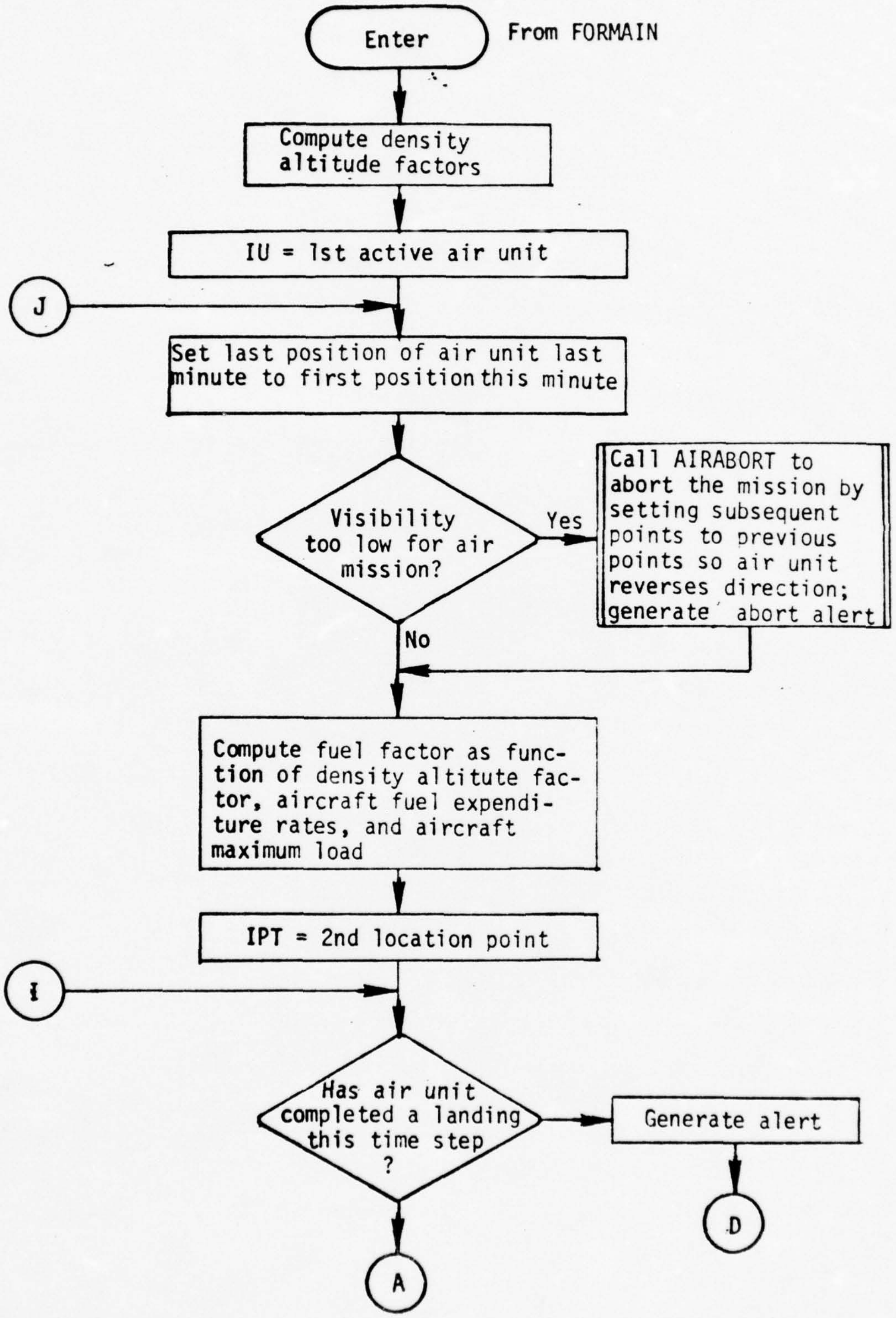

Enter
From FORMAIN
Compute density altitude factors
IU = 1st active air unit
J
Set last position of air unit last minute to first position this minute
Visibility too low for air mission?
Yes
Call AIRABORT to abort the mission by setting subsequent points to previous points so air unit reverses direction; generate abort alert
No
Compute fuel factor as function of density altitute factor, aircraft fuel expenditure rates, and aircraft maximum load
IPT = 2nd location point
I
Has air unit completed a landing this time step ?
Generate alert
D
A

Figure 5-119. General Logic Flow of AIRMOV Chain

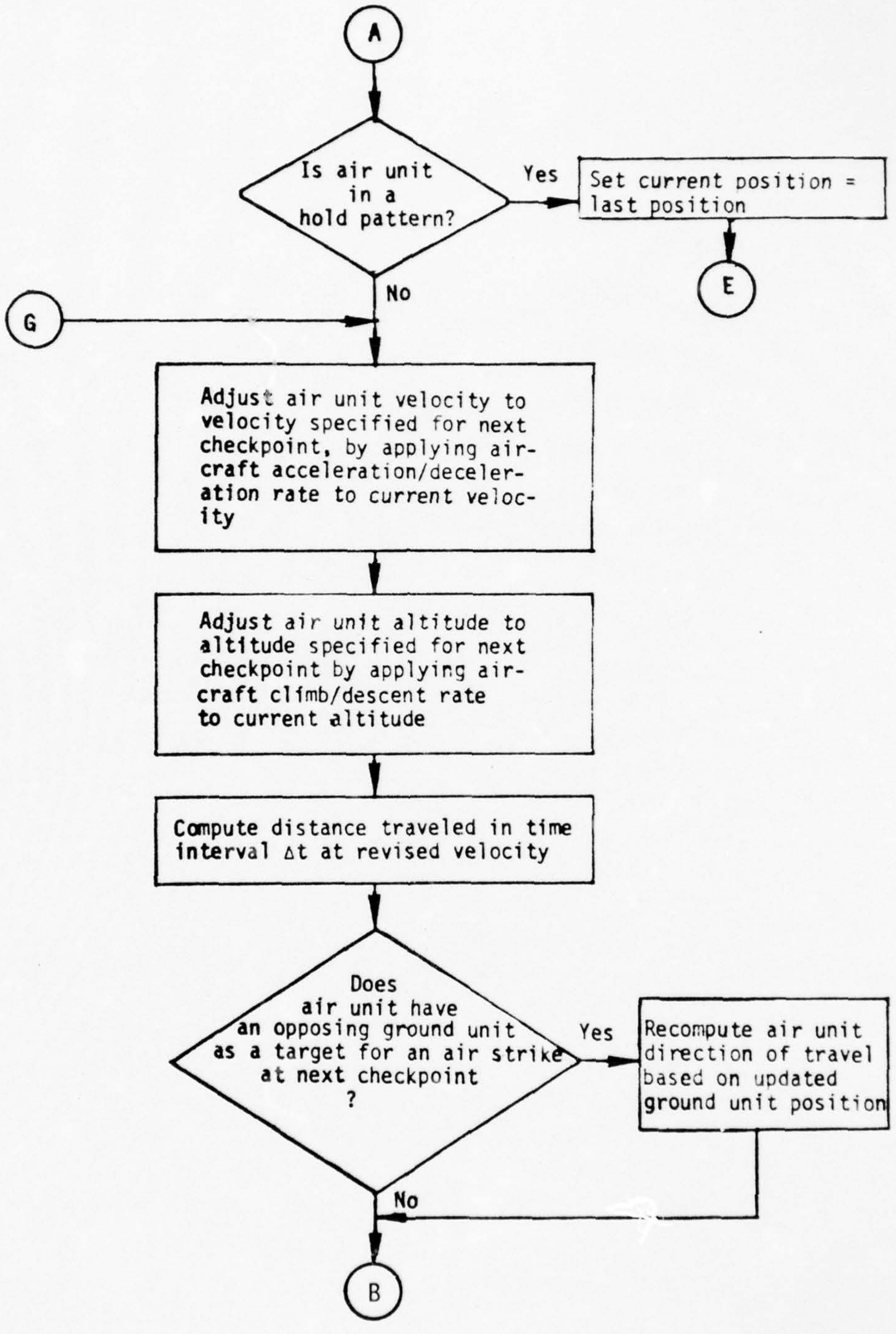

A
Is air unit in a hold pattern?
Yes
Set current position = last position
E
No
G
Adjust air unit velocity to velocity specified for next checkpoint, by applying aircraft acceleration/deceleration rate to current velocity
Adjust air unit altitude to altitude specified for next checkpoint by applying aircraft climb/descent rate to current altitude
Compute distance traveled in time interval Δt at revised velocity
Does air unit have an opposing ground unit as a target for an air strike at next checkpoint ?
Yes
Recompute air unit direction of travel based on updated ground unit position
No
B

Figure 5-119. General Logic Flow of AIRMOV Chain, Cont'd.

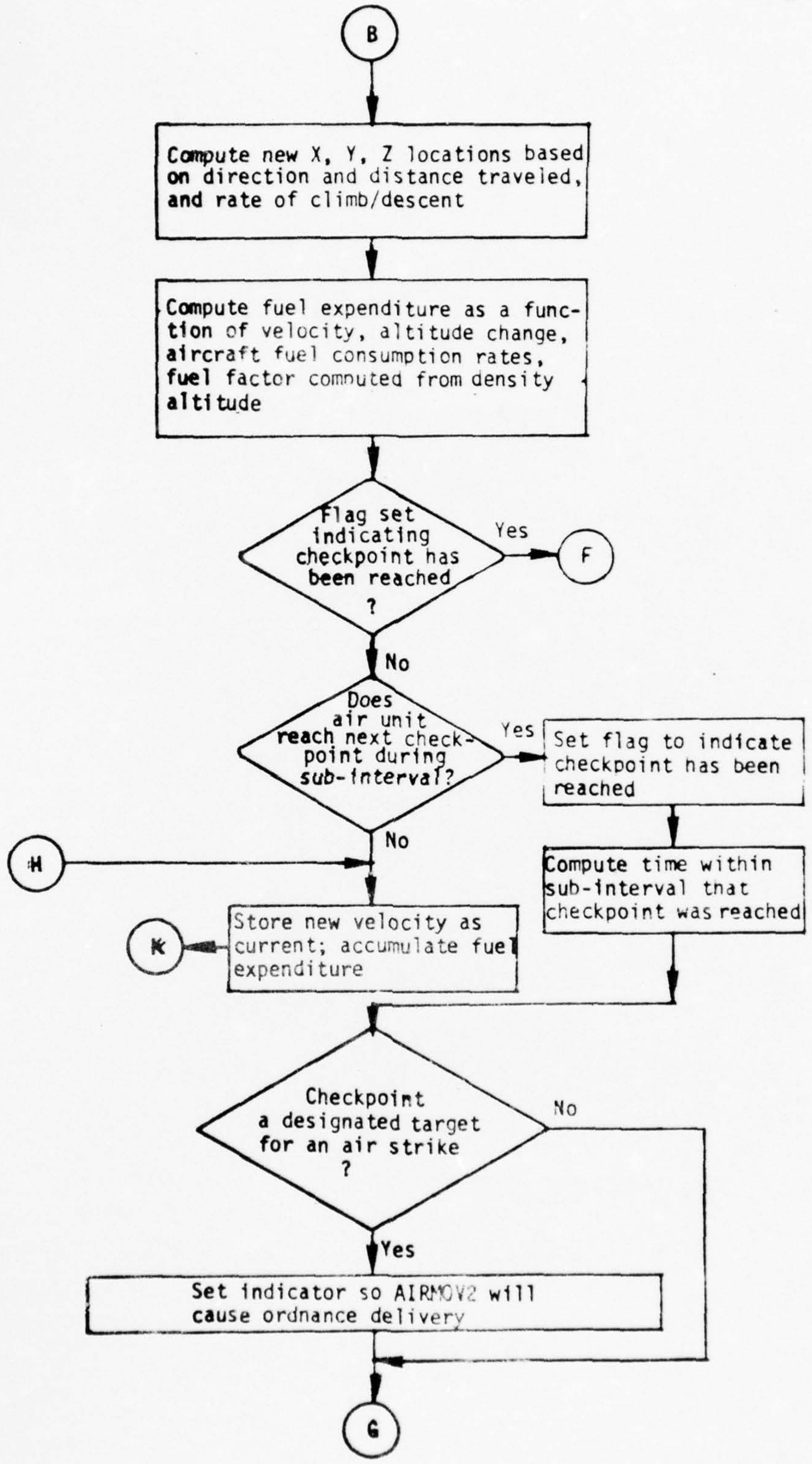

B
Compute new X, Y, Z locations based on direction and distance traveled, and rate of climb/descent
Compute fuel expenditure as a function of velocity, altitude change, aircraft fuel consumption rates, fuel factor computed from density altitude
Flag set indicating checkpoint has been reached ?
Yes
F
No
Does air unit reach next checkpoint during sub-interval?
Yes
Set flag to indicate checkpoint has been reached
Compute time within sub-interval that checkpoint was reached
No
H
Store new velocity as current; accumulate fuel expenditure
K
Checkpoint a designated target for an air strike ?
No
Yes
Set indicator so AIRMOV2 will cause ordnance delivery
G

Figure 5-119. General Logic Flow of AIRMOV Chain, Cont'd.

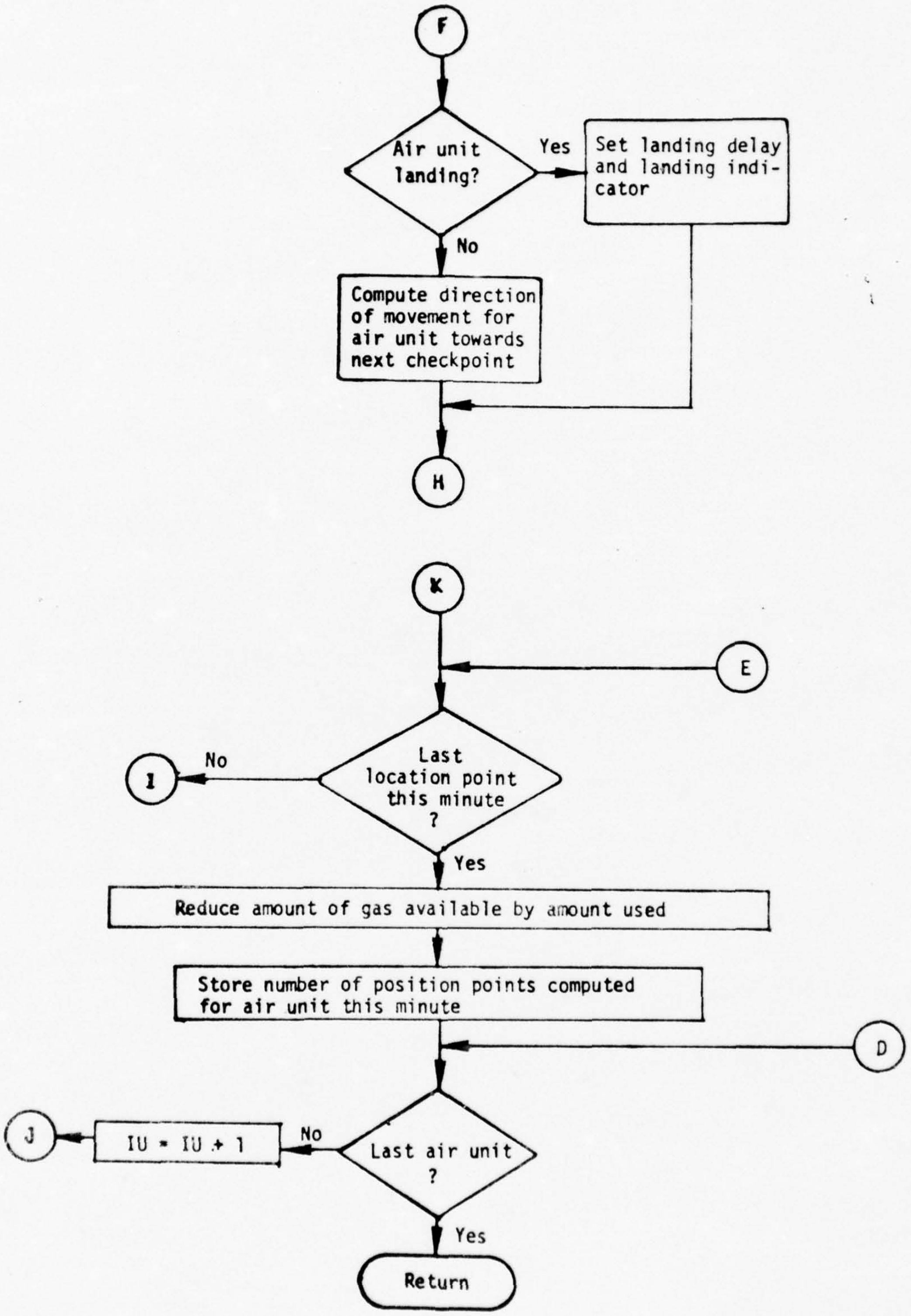

F
Air unit landing?
Yes
Set landing delay and landing indicator
No
Compute direction of movement for air unit towards next checkpoint
H
K
E
Last location point this minute ?
No
I
Yes
Reduce amount of gas available by amount used
Store number of position points computed for air unit this minute
D
Last air unit ?
No
IU = IU + 1
J
Yes
Return

Figure 5-119. General Logic Flow of AIRMOV, Cont'd.

Fuel consumption is a particularly complicated and non-linear function of a large number of factors, including air density, airspeed, aircraft gross weight, aircraft drag factor, air temperature and many more. In order to account for these factors and still provide an air module that is not far too complex for this application, a number of very significant simplifications have been made. The factors considered in the fuel consumption model are air density, aircraft load, climb/dive rate and aircraft speed. Fuel consumption has been assumed to be a linear, or in one case, a piece-wise linear, function of these four factors. The factors used to define the curves are aircraft type dependent and are input as a part of the equipment input deck. The definition of the factors and the corresponding variable names from the equipment input deck are shown below:

$$\text{ROFE(IAC,7)} - \frac{\text{Fuel consumption at worst air density}}{\text{Fuel consumption at best air density}} - 1$$

$$\text{ROFE(IAC,6)} - \frac{\text{Fuel consumption at maximum load}}{\text{Fuel consumption at minimum load}} - 1$$

ROFE(IAC,1) - Fuel consumption change for losing altitude

ROFE(IAC,2) - Fuel consumption change for gaining altitude

ROFE(IAC,3) - Fuel consumption rate at minimum speed

ROFE(IAC,4) - Fuel consumption rate at cruise speed

ROFE(IAC,5) - Fuel consumption rate at maximum speed

The actual consumption curves are shown in Figures 5-120, 5-121, 5-122, 5-123, 5-124 and 5-125. In the model the curve representing Figure 5-125 is calculated from the airspeed and Figure 5-124.

The subroutine descriptions for the subroutines called by AIRMOV are as follows: for AIRABORT and SWITCH see Section 5.8.1; and for ALERTGEN see Section 5.7.2.

5.8.1.3 AIRMOV2 Chain

AIRMOV2 is called once each time-step. It loops through each quarter minute subinterval to examine the location of each active air unit at the beginning and end of the subinterval. The AIRMOV2 chain subroutine linkage is shown in Figure 5-111. Table 5-58 presents a brief description of each subroutine in AIRMOV2, including the principal inputs and outputs. The general logic flow of AIRMOV2 is presented in Figure 5-126.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs		
FORMAIN	Secondary main program of the mathematical model. FORMAIN controls the minute-to-minute function :ow between modules and submodules of the mathematical model.	Input:	None	
		Output:	BDIR(JU,J)	- Sine and cosine of angle faced by unit JU during preceding time-step (cartesian coordinate system).
			BROMU(JU)	- Rate of movement of unit JU (in meters/minute for the preceding time-step.
			ITIME	- Current time (initial time at input).
			ITIMEC	- JTIMEO (in same common) minus ITIME. Used by subroutine FORSIG to calculate command and control event activation times (in minutes) since simulation start.
			JTIMEO	- Clock time (in minutes) corresponding to start of simulation.
			KTIME	- Time-step number.
			NDAYE	- Calendar date of first day of simulation. The lowest numbered NDAYE for a set of exercises should be used as the first day of the set and used to set NDAYEMN1.
			NHOURE	- Number of elapsed hours since midnight (0-23).
			NMINE	- Number of elapsed minutes (0-59).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
FORMAIN (Cont'd)		TIME - Time (in floating point minutes).
AIRMOV2	Computes the results of the aircraft's flight over the locations calculated by AIRMOV.	Input: DCAS - Number of casualties inflicted by weapon.
		IACPX(JCKPT,JAU) - X-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \le JCKPT \le 9$, $1 \le JAU \le 10$)
		IACPY(JCKPT,JAU) - Y-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \le JCKPT \le 9$, $1 \le JAU \le 10$)
		IAIRDFLG(JU) - Air defense flag for ground unit JU. ($1 \le JU \le 100$) = 1 Fire at will = 2 Fire only if attacked = 3 Do not fire
		JTGTPT(JAU) - Index within/AIRLOC/arrays for target point for air unit JAU. ($1 \le JAU \le 100$) = 0 If air unit JAU has no target during this time-step.
		NODETECT - = TRUE if air unit was not detected at any point in this quarter minute. = FALSE otherwise.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRMOV2 (Cont'd)		NTGTYP(JAU) - Type of target for air unit JAU. ($1 \leq JAU \leq 100$) = Ground unit number or code for road, bridge, or X-Y point. Output: IXYIM(JU) - Each element of this array, if non-zero, indicates the X-Y location of an impacting fire, as follows: - Sign of the word indicates red (-) or blue (+) - After converting to a positive number, bits 0-15 - - X location bits 16-31 - - Y location ($1 \leq JU \leq 100$) PERS(JU) - Number of personnel currently in unit JU. ($1 \leq JU \leq 100$) TOTEQU(IU,J) - Total number of pieces remaining of J-th equipment type carried by unit IU. ($1 \leq IU \leq 100$, $1 \leq J \leq 14$) USEEQU(JU,J) - Number of pieces manned for the J-th equipment type carried by unit JU. ($1 \leq JU \leq 100$, $1 \leq J \leq 14$)
GRNDAIR	Determines if ground unit IGU detected air unit IAU this quarter minute.	Input: ALL - Global ambient light level (in foot lamberts). LLLN - Maximum light level (in foot lamberts) for using any night optical device.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
GRNDAIR (Cont'd)		VISM - Global meteorological visibility (in meters).
		Output: DTCTATPT(JAU) - = TRUE if air unit was detected at point JAU. (JAU is index to/AIRLOC/ arrays.) (1≤JAU≤8)
		IGADET(JU,JAU) - Bit matrix indicating whether ground unit IGU has already detected air unit JAU. IGU is used to determine JU. (1≤JU≤4, 1≤JAU≤10)
		NODETECT - (See subroutine AIRMOV2.)
AIRCAS	Determines amount of fire and resulting *casualties*.	*Input:* DTCTATPT(JAU) - (See subroutine GRNDAIR.)
		I - Index to the firing unit in firing routines.
		IFEF - Weapon effects function number.
		II - Number of air unit. Used by subroutines ORDPRI and AIREVENT; used as target unit by firing routines. (NFDU≤II≤NLDU)
		IPVCE(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ)≥0), personnel vulnerability class associated with each mode IMODE of equipment type IEQ. For air equipment other than an aircraft (IEQCOD(IEQ)=-2 or -3), equipment number of the IMODE-th allowable aircraft *type on which* IEQ may be used. (Hence, a maximum

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRCAS (Cont'd)		of eight aircraft per equipment type.) The first zero encountered implies no other allowable aircraft. (1≤IEQ≤80, 1≤IMODE≤8) IXAIR(JAIRTE,JAU) - X-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPX). (1≤JAIRTE≤8, 1≤JAU≤10) IXY(IU,J) - Actual position coordinates (J=1 for X, J=2 for Y) of unit IU. (1≤IU≤100) IYAIR(JAIRTE,JAU) - Y-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPY). (1≤JAIRTE≤8, 1≤JAU≤10) IZAIR(JAIRTE,JAU) - Z-coordinate of air unit JAU at JAIRTE-th point in this minute. There are 4 quarter-minute points in current minute plus 0 to 4 intermediate check points (IACPZ). (1≤JAIRTE≤8, 1≤JAU≤10) ROFE(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ)≥0), rate of fire of weapon type IEQ in each mode IMODE (in rounds/minute).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRCAS (Cont'd)		For aircraft (IEQCOD(IDQ)=-1), IMODE = 1 Fuel expenditure for losing altitude (in pounds/meter). = 2 Fuel expenditure for gaining altitude (in pounds/meter). = 3 Fuel expenditure at minimum speed, minimum load, best pressure density (in pounds/minute). = 4 Fuel expenditure at cruise speed (in pounds/minute). = 5 Fuel expenditure at maximum speed (in pounds/minute). = 6 Ratio of fuel expenditure at maximum load to fuel expenditure rate at minimum load. = 7 Ratio of fuel expenditure rate at worst pressure density to fuel expenditure at best pressure density. = 8 Not used. For air ordnance (IEQCOD(IEQ)=-2),

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRCAS (Cont'd)		IMODE = 1 Fraction of personnel vulnerability class 1 (standing) and within target area who are killed by this equipment. = 2 Fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equipment. = 3 Fraction of personnel vulnerability class 3 (prone) and within target area who are killed by this equipment. = 4 Fraction of equipment with IEQCLS=1 and within target area which is damaged by this equipment. = 5 Fraction of equipment with IEQCLS=2 and within target area which is damaged by this equipment. = 6 Fraction of equipment with IEQCLS=3 and within target area which is damaged by this equipment.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRCAS (Cont'd)		= 7 Fraction of equipment with IEQCLS=4 and within target area which is damaged by this equipment. = 8 Fraction of equipment with IEQCLS=5 and within target area which is damaged by this equipment. (1≤IEQ≤80, 1≤IMODE≤8) SIGU(JU) - Suppression factor: fraction of men in unit JU who are suppressed. (1≤JU≤100) USEEQU(JU,J) - (See subroutine AIRMOV2.) Output: DCAS - (See subroutine AIRMOV2.) FIRE - Volume of fire (in rounds/unit time). FIRE(JU) - = TRUE if ground unit JU has already fired its air defense weapons during current quarter minute. (1≤JU≤100) IAMT(JTGT) - Number of rounds fired at the JTGT-th target of a given weapon. (1≤JTGT≤50) IMODE(JTGT) - For support fire weapons, the mode of fire (1 to 8) used of a given support fire weapon; for air defense weapons, the quarter minute

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRCAS (Cont'd)		(1 to 4) during which the JTGT-th air unit was fired at by a given air defense weapon; for other weapons, not used. ($1 \le JTGT \le 50$) ITRGT(JTGT) - Number of JTGT-th target unit being fired at by a given weapon. ($1 \le JTGT \le 50$) PERNVC(JU,IPVC,K) - Number of personnel in in each vulnerability class IPVC for unit JU where: K = 1 Actual number in previous time-step. = 2 Actual number in current *time-step*. = 3 Unsuppressed number in previous time-step. = 4 Unsuppressed number in current time-step. ($1 \le IPVC \le 6$, $1 \le JU \le 100$)
ADFALERT	Generates appropriate air defense alerts.	Input: IFCODE - = 0 Ground unit no longer firing at air unit. = 1 Ground unit has opened fire on air unit. IPRVENG(JU,JAU) - Bit matrix indicating whether ground unit IGU is currently engaging air unit JAU with air defense weapons. IGU is used to determine JU. ($1 \le JU \le 4$, $1 \le JAU \le 10$)

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
ADFALERT (Cont'd)		Outnut: IPRVENG(JU,JAU) -(See "Input" for this subroutine.) (Also outputs air defense fire engagement alerts.)
AIRGRND	Determines if air unit IAU detected ground unit IGU this quarter minute.	Input: IEQCOD(IEQ) - Equipment category code of a given equipment type IEQ. ($1 \leq IEQ \leq 80$) = -3 Air sensor = -2 Air weapon = -1 Aircraft = 0 Not a weapon = 1 Direct fire weapon = 2 Indirect fire weapon = 3 Support fire weapon = 4 Air defense weapon IMAXRE(IEQ,ITGTE) - For ground equipments (indicated by IEQCOD(IEQ)≥0), maximum firing range (in meters) of each weapon type IEQ against primary (ITGTE=11), secondary (ITGTE=2) target elements for air equipment other than an aircraft (IEQCOD(IEQ)=-2 or -3), maximum range at which equipment may be used (ITGTE=1). ($1 \leq IEQ \leq 80$) IMINRE(IEQ) - For ground equipments (indicated by IEQCOD(IEQ)≥0), minimum firing range (in meters) for weapon (equipment) type

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRGRND (Cont'd)		IEQ. For air equipment other than an aircraft (IEQCOD(IEQ)= -2 or -3), minimum range at which equipment may be used. $(1 \leq IEQ \leq 80)$ IXAIR(JAIRTE,JAU) - (See subroutine AIRCAS.) IXY(IU,J) - (See subroutine AIRCAS.) IYAIR(JAIRTE,JAU) - (See subroutine AIRCAS.) IZAIR(JAIRTE,JAU) - (See subroutine AIRCAS.) ROME(IEQ,IMODE) - For ground equipments (indicated by $IEQCOD(IEQ) \geq 0$), rate of movement (in meters/minute) of weapon type I in mode IMODE (unobstructed). For aircraft (IEQCOD(IEQ)=-1), IMODE = 1 Maximum load aircraft can carry (in pounds) at best modeled pressure density (PDBEST). = 2 Maximum altitude of aircraft (in meters). = 3 Minimum speed of aircraft (in meters/minute). = 4 Cruise speed of aircraft (in meters/minute). = 5 Maximum speed of aircraft (in meters/minute).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRGRND (Cont'd)		= 6 Maximum load aircraft can carry (in pounds) at worst modeled pressure density (PDWORST); the correct negative value will insure that an aircraft cannot fly at pressure densities below its capability. = 7 Poorest meteorological visibility in which aircraft can continue its mission (in meters). = 8 Not used. For equipment other than aircraft (IEQCOD(IEQ)=-2 or -3), IMODE = 1 Weight of equipment (in pounds) including standard ammunition load. = 2 Minimum speed (in meters/minute) at which equipment can be used. = 3 Maximum speed (in meters/minute) at which equipment can be used. = 4 Minimum altitude at which equipment can be used (in meters).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
AIRGRND (Cont'd)		= 5 Maximum altitude at which equipment can be used (in meters). = 6 For sensors, not used; for ordnance, road crater radius (in meters) against road type 1. = 7 For sensors, not used; for ordnance, road crater radius (in meters) against road type 2. = 8 For sensors, not used; for ordnance, road crater radius (in meters) against road type 3. ($1 \leq IEQ \leq 80$, $1 \leq IMODE \leq 8$) ROMU(JU) - Rate of movement of unit JU (in meters/minute). ($1 \leq JU \leq 100$) USEEQU(JU,J) - (See subroutine AIRMOV2.)
ADW	Computes ground casualties resulting from delivery of air ordnance.	Input: DERROR - Ordnance delivery ballistic error perpendicular to the direction of flight of the air unit. IAU - Number of air unit. ($NFDU \leq IAU \leq NLDU$) NIMFIRES - Size of IXYIM array.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
ADW (Cont'd)		NMPN(JAU) - *Ordnance type being delivered* by air unit JAU. (JAU=1, 10) NQPTOT(JCOLOR) - Number of equipment types for all red (JCOLOR=1), and blue (JCOLOR=2) units used for *stats arrays.* PTLNTH - Length (in meters) of lethal area caused by air ordnance impact. *PTWDTH - Width (in meters)* of lethal area caused by air ordnance impact. RERROR - Ordnance delivery ballistic *error in the direction of flight of the air unit.* Output: DNU(JU,J) - Casualties for equipment J in unit JU during current time-step. ($1 \le JU \le 100$, $1 \le J \le 14$) DPERS(JU) - Personnel casualties for unit JU during current time-step. ($1 \le JU \le 100$) IXYIM(JU) - (See subroutine AIRMOV2.) IXYIMPTR(JMPF) - Pointer into IXYIM array used by air units delivering ordnance, enabling a special symbol to be drawn for air impacting fires. ($1 \le JMPF \le 10$) RELCAS - Number of equipment-related personnel casualties.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
FINFUN	Determines the unique weapon effects entry with which to compute casualty assessment (this entry determines the weapon effects functions and coefficients to use in calculating the function).	Input: ICL - Personnel vulnerability class of target equipment number. IEQLM(I) - Begin (I=1) and end (I=2) of weapon effects function table search (table is sorted on equipment number). IFNTB(I,J) - Weapon effects look-up table is of the form SSXXYZZFADDD, where IFNTB(I,1) = SSXXYZZ IFNTB(I,2) = FADDD I=I-th ($1 \leq I \leq 1000$) entry in the weapon effects look-up table such that SS = Operational state of target unit (00 ≥ any state) XX = Weapon type Y = Mode of fire (0 ≥ any mode) ZZ = Target personnel vulnerability class (1-6) (see variable NPFC) or target equipment number (1-80) or target bridge type 91 = bridge type 1 92 = bridge type 2 or target road type 95 = road type 1 96 = road type 2

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
FINFUN (Cont'd)		97 = road type 3 F = Weapon effects function to be used. A = Allocation criterion for aimed fire versus equipment. DDD = Number of associated grouping of constants to be used with effects function (1-315). Table must be all entries against personnel first, with all artillery versus personnel grouped at the end of the group; then all entries against equipment, roads or bridges, with the anti-aircraft versus air equipment grouped at the end of this group. IPLM(I) - Begin (I=1) and end (I=2) of weapon effects function table search against personnel as target. Output: IAEF - Allocation criterion for aimed fire versus equipment. IDEF - Number of entry in EFFDAT table. IFEF - Effects function number.
EFFNS	Computes casualties based on the selected effects function.	Input: EFFDAT(ICOEF,J) - Groupings of four constants (J) to be used with the ICOEF-th effects functions. ($1 \le ICOEF \le 315$)

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
EFFNS (Cont'd)		EFFRAC - Fraction of rounds fired at target area that fall within that area. FIRE - (See subroutine AIRCAS.) PERNVC(JU,IPVC,K) - (See subroutine AIRCAS.) Output: DCAS - (See subroutine AIRMOV2.) RELCAS - (See subroutine ADW.)
ADWDATA	Computes average delivery errors from ordnance type, velocity, and altitude.	Input: IACPV(JCKPT,JAU - Velocity (in meters/minute) at JCKPT-th check point for air unit JAU. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 9$) IACPZ(JCKPT,JAU) - Z-coordinate of JCKPT-th check point for air unit JAU. Check points are the points put in using the air menu. ($1 \leq JCKPT \leq 9$, $1 \leq JAU \leq 9$) IEQCLS(IEQ) - For ground equipments (indicated by IEQCOD(IEQ)$\geq$0), for each ground equipment type IEQ, a general classification for air ordnance, defined as follows: 0 = No casualty effect is computed due to air ordnance for ground equipment. 1 = Small arms 2 = Light mortars 3 = Artillery 4 = Non-armored vehicles

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
ADWDATA (Cont'd)		5 = Armored vehicles For air ordnance (IEQCLS(IEQ)= -2), IEQCLS = 1 IEQ is a 250-lb. low-drag bomb. = 2 IEQ is a 500-lb. low-drag bomb. = 3 IEQ is a 750-lb. low-drag bomb. = 4 IEQ is a 1000-lb. low-drag bomb. = 5 IEQ is a 2000-lb. low-drag bomb. = 6 IEQ is a 250-lb. high-drag bomb. = 7 IEQ is a 500-lb. high-drag bomb. = 8 IEQ is a 750-lb. high-drag bomb. = 9 IEQ is a Maverick. = 10 IEQ is a Shrike. = 11 IEQ is 2.75-inch rockets. = 12 IEQ is 20-MM cannon. = 13 IEQ is CBU-2. = 14 IEQ is CBU-24. = 15 IEQ is Rockeye. = 16 IEQ is Napalm. Not used for aircraft.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs		
ADWDATA (Cont'd)				($1 \leq IEQ \leq 80$)
		Output:	DERROR	- (See subroutine ADW.)
			PTLNTH	- (See subroutine ADW.)
			PTWDTH	- (See subroutine ADW.)
			RERROR	- (See subroutine ADW.)
NORMAL	Calculates a normally distributed random number with specified mean and standard deviation.	Input:	XMEAN	- Specified mean.
			XSTDEV	- Specified standard deviation.
		Output:	RVNORM	- Normally distributed random variable.
FRFD	Computes the fraction of a target unit overlapping with the lethal area of an air-delivered *weapon*.	Input:	ID	- Depth of unit being tested (in meters).
			IW	- Width of unit being tested (in meters).
			IXBOUND(I)	- Values for X (I=1 or 2) bounding the lethal area of impacting ordnance (IXBOUND(2) is the smaller value).
			IX1	- X-coordinate of unit being tested less half the unit width IW.
			IX2	- Same as IX1 except plus IW/2.
			IYBOUND(I)	- Values for Y (I=1 or 2) bounding the lethal area of impacting ordnance (IYBOUND(2) is the smaller value).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
FRFD (Cont'd)		IY1 - Y-coordinate of unit being tested less ID/2. IY2 - Same as IY1 except plus ID/2. Output: OVRLAP - Fraction of unit area overlapped by air ordnance lethal area.
OTHRDMG	Computes damage to bridges or roads near impact point of air-delivered weapon.	Input: DCAS - (See subroutine AIRMOV2.) IAMTE(IEQ,IMODE) - For ground equipments (indicated by IEQCOD(IEQ)≥0), ammunition type used by weapon type IEQ in mode of operation IMODE (0 implies no ammunition used). For aircraft (IEQCOD (IAC)=-1), IMODE = 1 Takeoff delay time for this aircraft (in minutes). = 2 Landing delay time for this aircraft (in minutes). For air ordnance (IEQCOD(IAC)= -2), IMODE = 1 Number of ammunition type for this equipment, if any. = 2 Number of rounds in a standard load of that ammunition type. = 3 Rate of fire of this equipment (in rounds/minute).

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs
OTHRDMG (Cont'd)		= 4 Number of drops per pass (applies to bombs). = 5 Distance between drops (in meters). = 6 Dud probability times 100. = 7 Kill probability times 100 for bridge type. = 8 Kill probability times 100 for bridge type. (1≤IEQ≤80, 1≤IMODE≤8) I1 - X-coordinate of ordnance impact point. I2 - Y-coordinate of ordnance impact point. I3 } I4 } Subscripts of IXAIR, IYAIR tables which give current air unit location. Output: BRGDMGE(JBR) - Fraction of bridge JBR damaged. (1≤JBR≤16) RDDMGE(JRDSG) - Fraction of road segment JRDSG damaged by fire. (1≤JRDSG≤500)
DIDITHIT	Determines if a bridge or road segment was damaged by an air-delivered weapon.	Input: J1 } J2 } (X, Y) coordinates of first point of bridge or road segment to be examined.

Table 5-58. AIRMOV2 Chain Subroutine Descriptions (Cont'd)

Subroutine	Description	Principal Inputs/Outputs		
DIDITHIT (Cont'd)			J3, J4	(X, Y) coordinates of second . point of bridge or road segment to be examined.
			J5, J6	(X, Y) coordinates of ordnance impact point.
			KK	- = 0 This is a bridge segment. ≠ 0 This is a road segment.
			KKK	- Bridge or road number.
		Output:	DCAS	- (See subroutine AIRMOV2.)
			K	- = 0 Segment not damaged. = 1 Segment was damaged.
ADWALRT	Generates an alert message for a ground unit taking casualties as a result of an air strike (air-delivered weapons) against it.	Input:	DNU(JU,J)	- Casualties for equipment J in unit JU during current time step. ($1 \leq JU \leq 100$, $1 \leq J \leq 14$)
			DPERS(JU)	- Personnel casualties for unit JU during current time-step. ($1 \leq JU \leq 100$)
		Output:	(Generates alerts for ground units taking casualties from air strike.)	

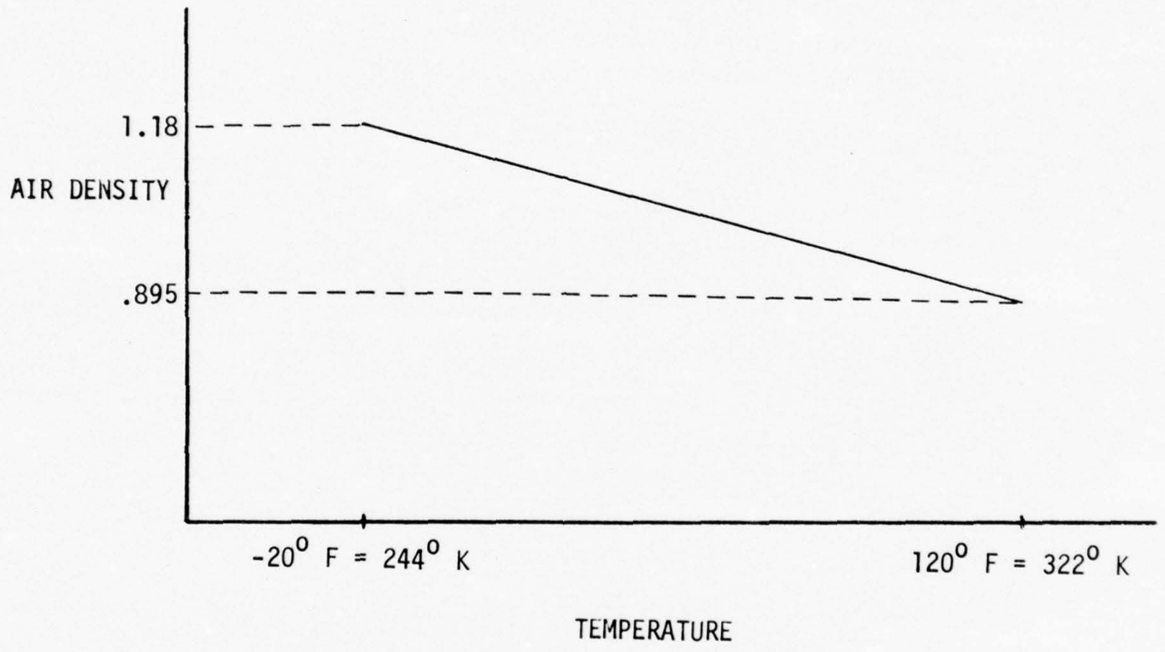


Figure 5-120. Temperature versus Air Density

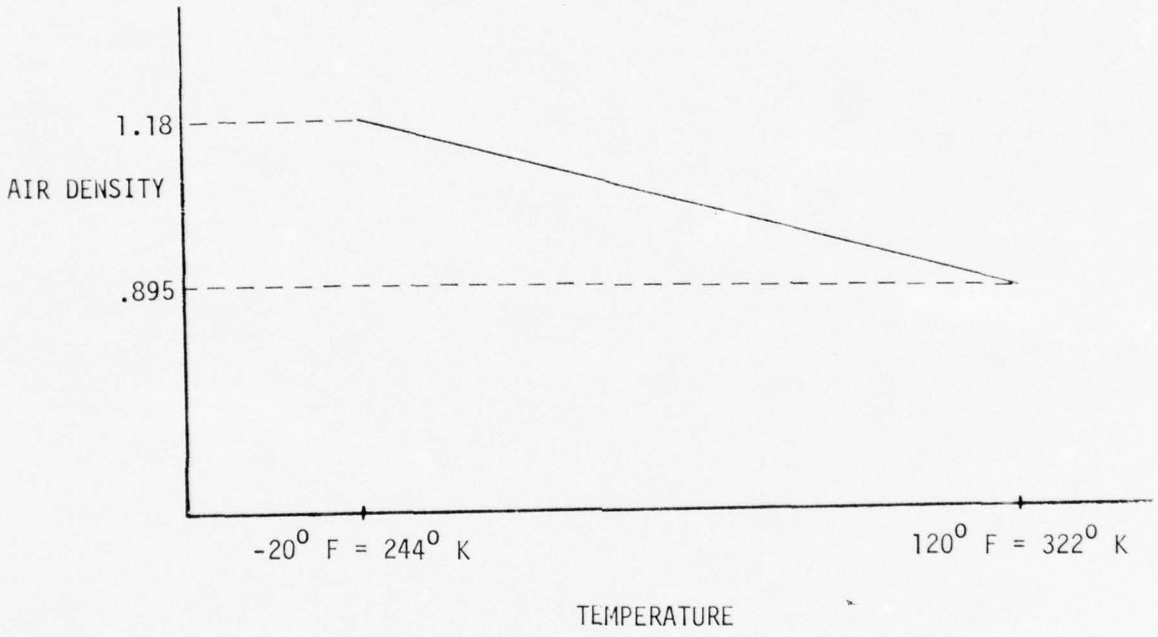


Figure 5-120. Temperature versus Air Density

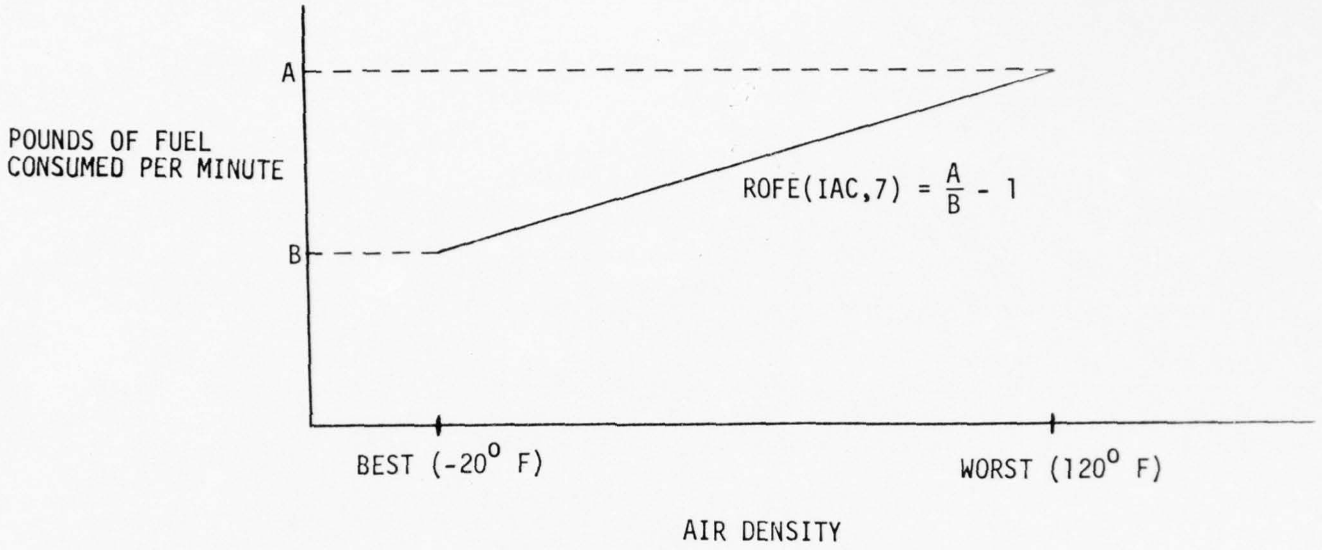


Figure 5-121. Aircraft Fuel Consumption versus Air Density (Temperature Dependent)

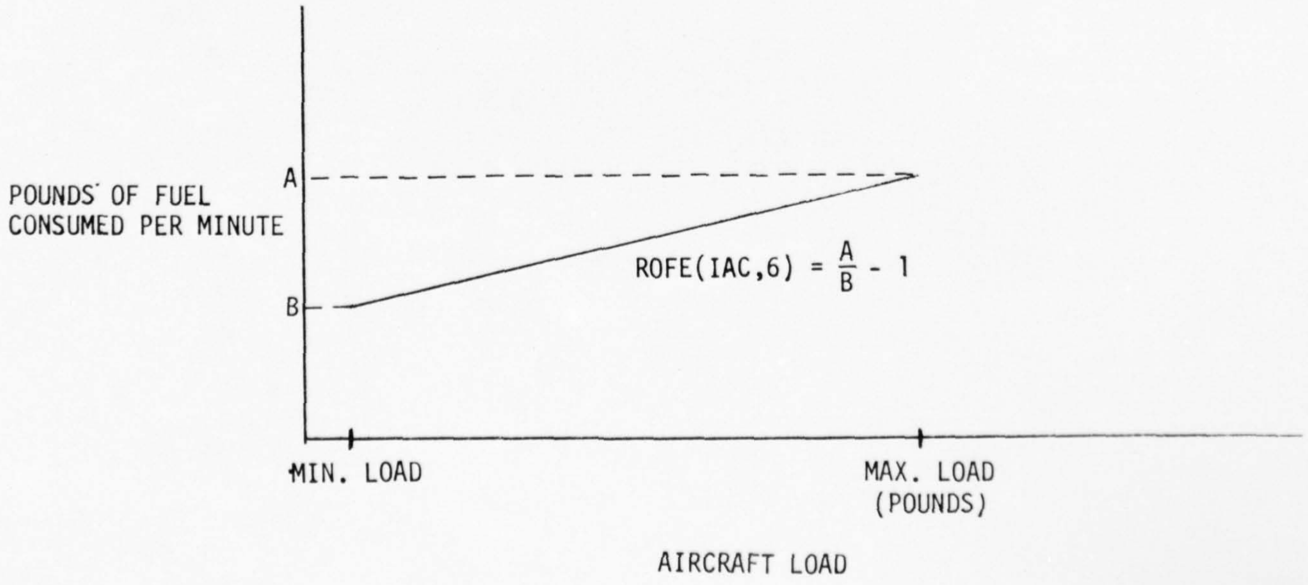


Figure 5-122. Aircraft Fuel Consumption versus Aircraft Load

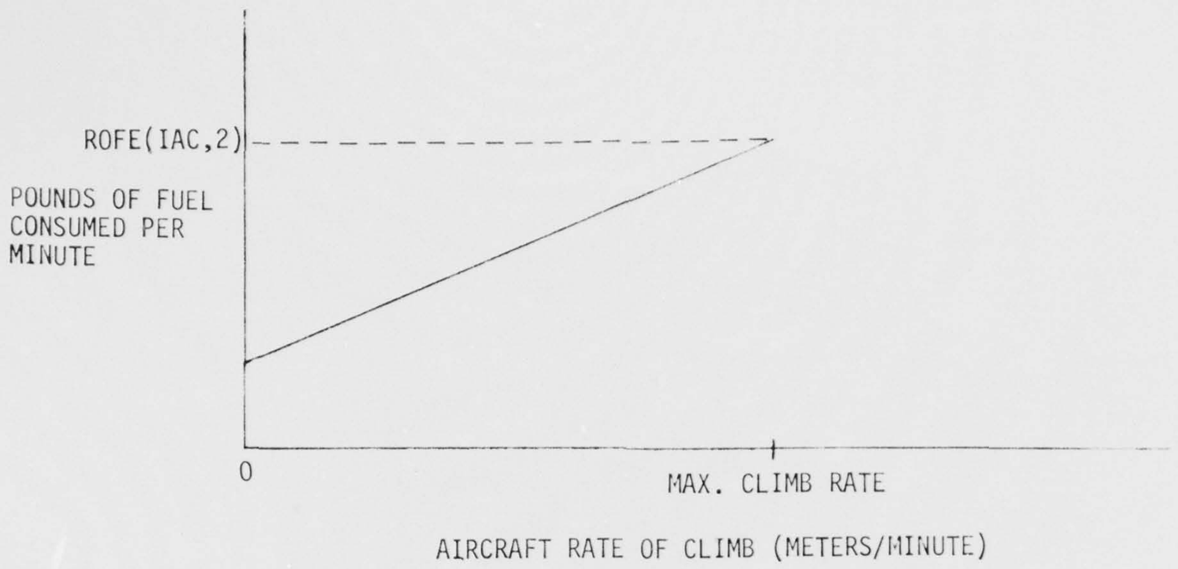


Figure 5-123. Aircraft Climb Rate versus Fuel Consumption

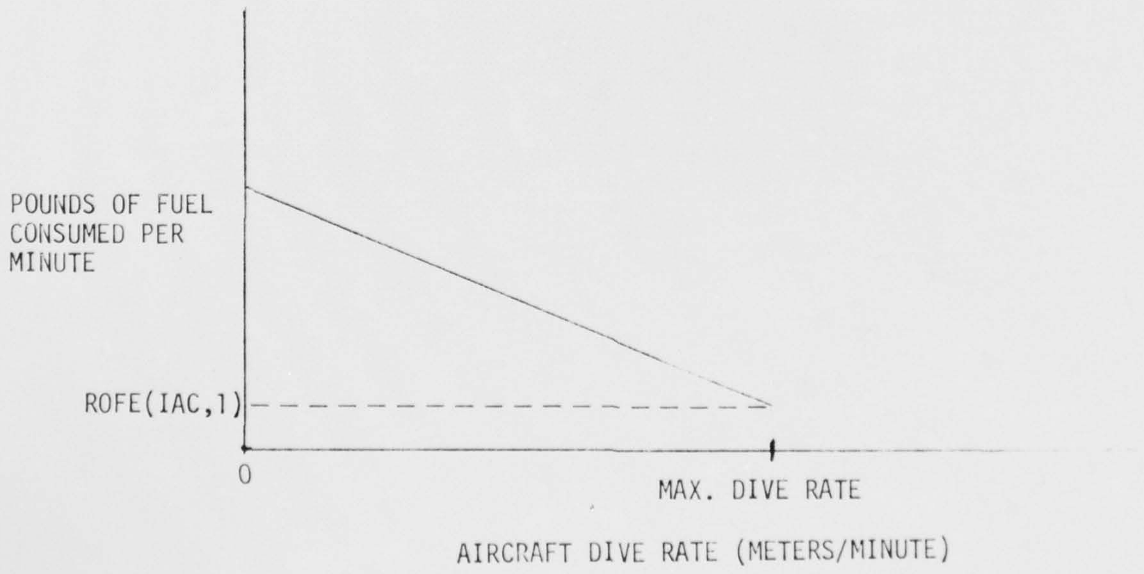


Figure 5-124. Aircraft Dive Rate versus Fuel Consumption

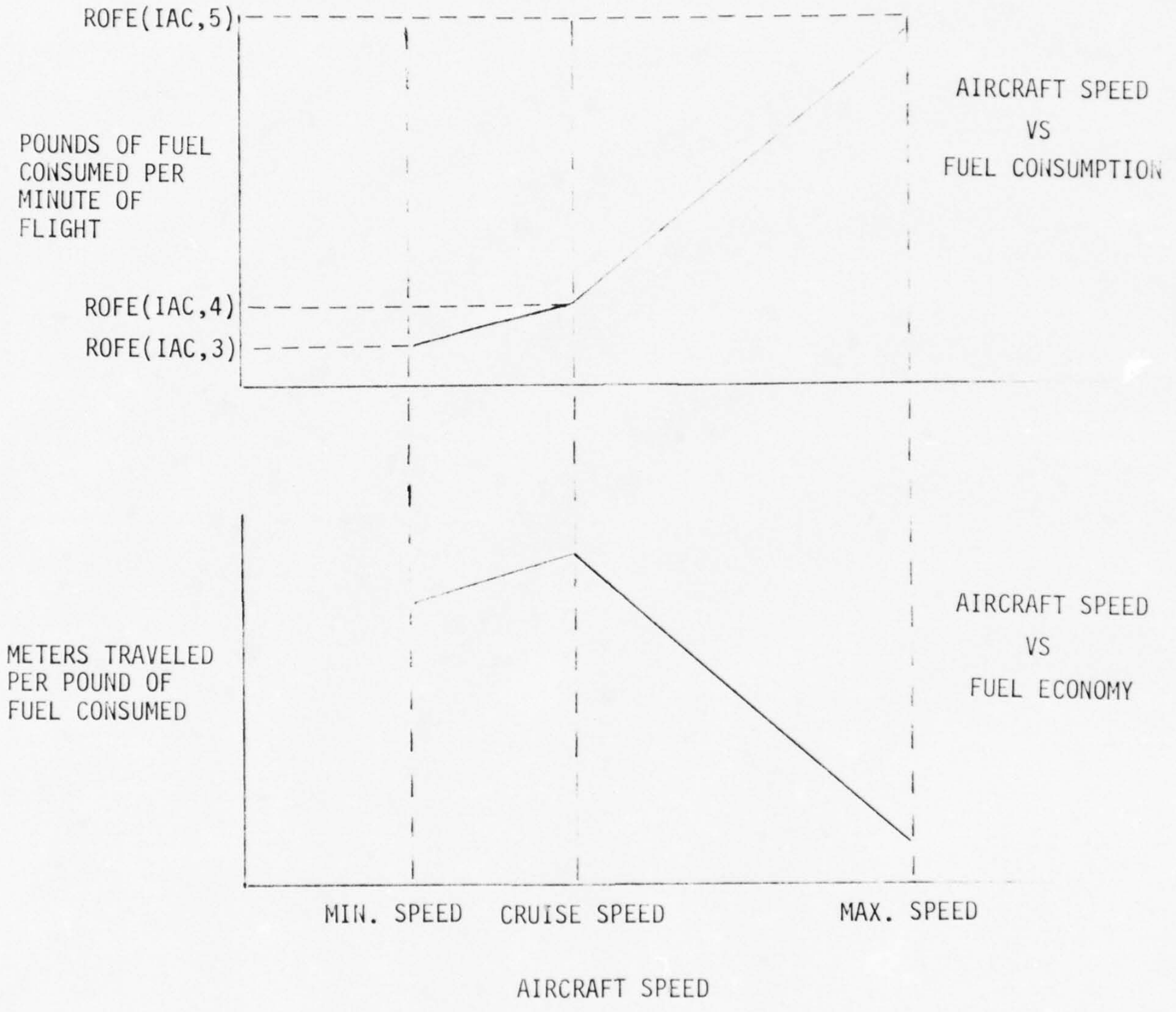


Figure 5-125. Calculations of Aircraft Fuel Consumption as a Function of Aircraft Speed

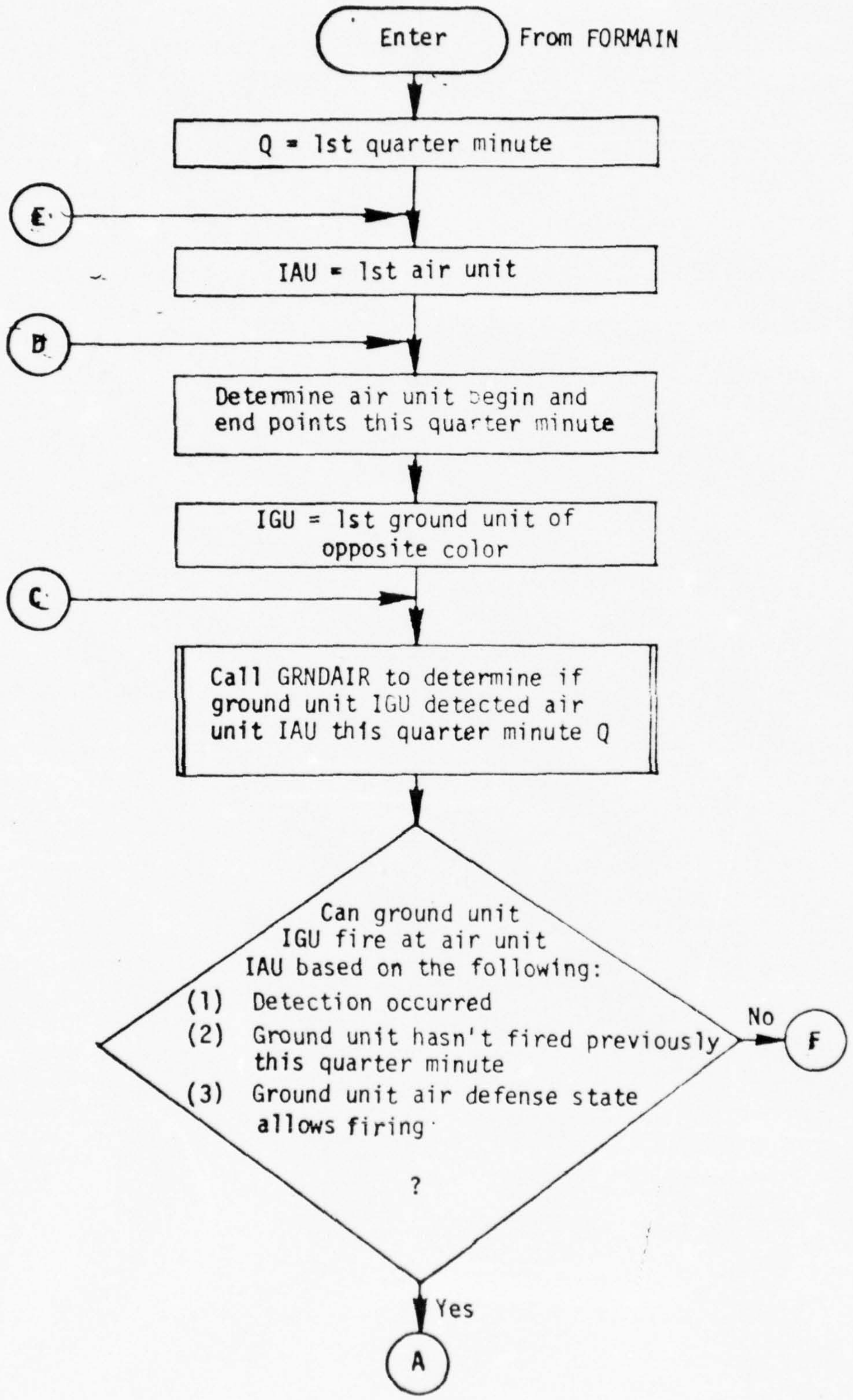

Enter
From FORMAIN
Q = 1st quarter minute
E
IAU = 1st air unit
D
Determine air unit begin and end points this quarter minute
IGU = 1st ground unit of opposite color
C
Call GRNDAIR to determine if ground unit IGU detected air unit IAU this quarter minute Q
Can ground unit IGU fire at air unit IAU based on the following:
(1) Detection occurred
(2) Ground unit hasn't fired previously this quarter minute
(3) Ground unit air defense state allows firing
?
No
F
Yes
A

Figure 5-126. General Logic Flow of AIRMOV2 Chain

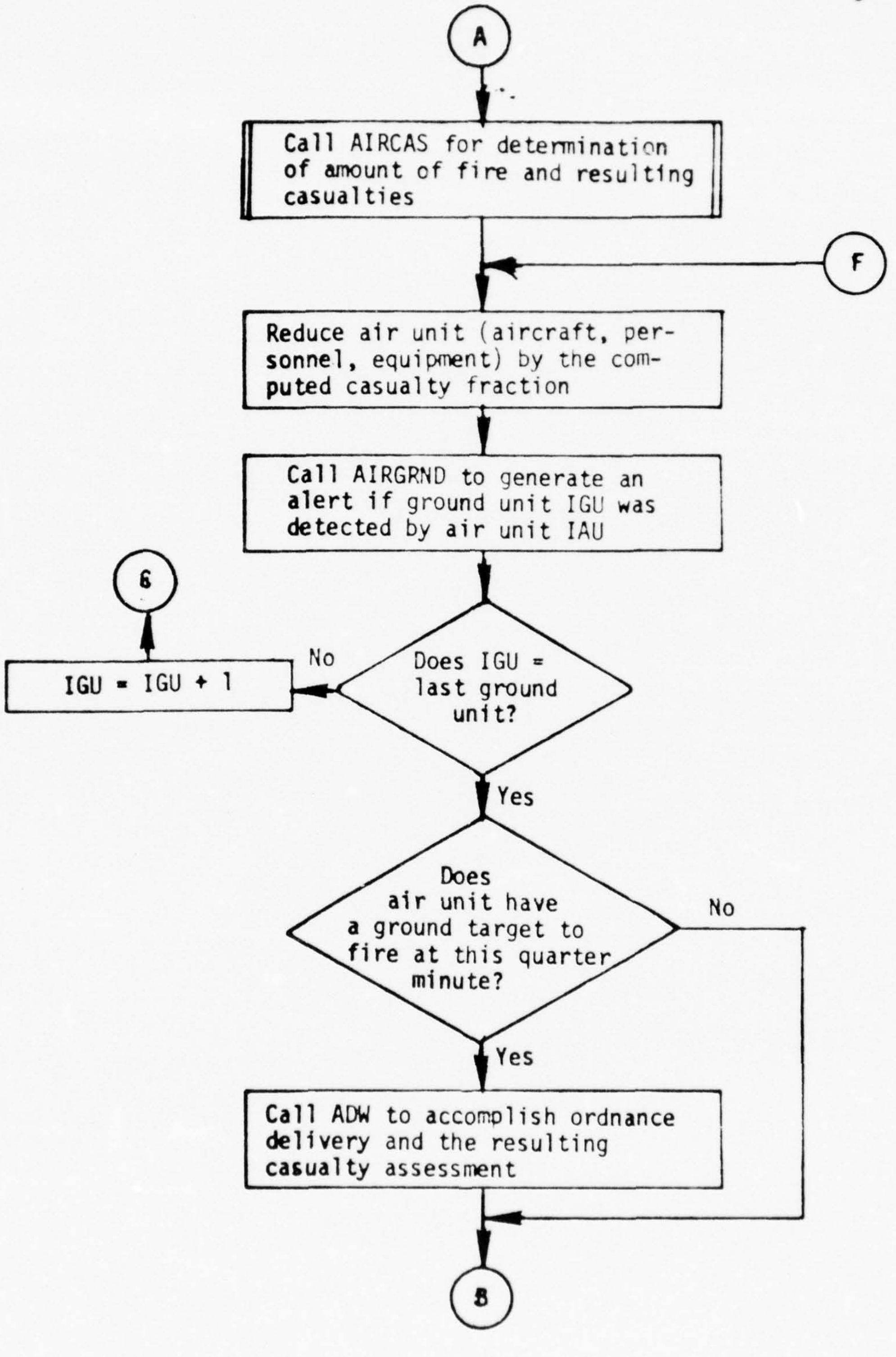

A
Call AIRCAS for determination of amount of fire and resulting casualties
F
Reduce air unit (aircraft, personnel, equipment) by the computed casualty fraction
Call AIRGRND to generate an alert if ground unit IGU was detected by air unit IAU
G
IGU = IGU + 1
No
Does IGU = last ground unit?
Yes
Does air unit have a ground target to fire at this quarter minute?
No
Yes
Call ADW to accomplish ordnance delivery and the resulting casualty assessment
B

Figure 5-126. General Logic Flow of AIRMOV2 Chain, Cont'd.

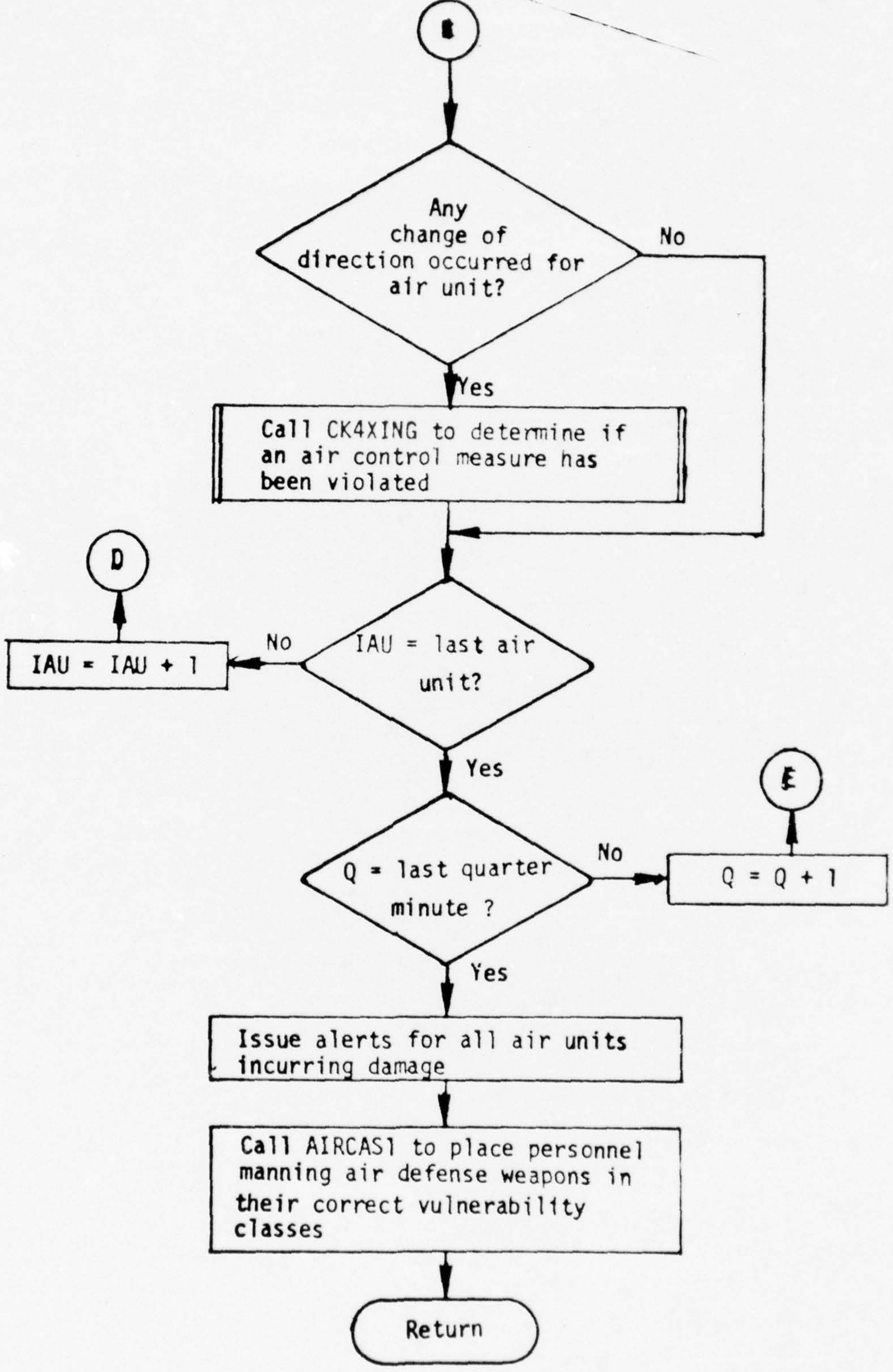

E
Any change of direction occurred for air unit?
No
Yes
Call CK4XING to determine if an air control measure has been violated
D
IAU = IAU + 1
No
IAU = last air unit?
Yes
E
Q = last quarter minute ?
No
Q = Q + 1
Yes
Issue alerts for all air units incurring damage
Call AIRCAS1 to place personnel manning air defense weapons in their correct vulnerability classes
Return

Figure 5-126. General Logic Flow of AIRMOV2 Chain, Cont'd.

AIRMOV2 checks all ground units of the opposite color to determine if detection by the ground unit has taken place (via a call to GRNDAIR). (For a description of GRNDAIR, see Section 5.3.2.) If so, the ground unit will consider firing at the air unit (via a call to AIRCAS) if the ground unit has not already fired at another air unit and it is in an air defense state which allows it to fire at the opposing air unit. If any aircraft are lost from the air defense fire, the number of aircraft lost and a proportionate number of personnel and equipment are removed from the air unit. Detection by the air unit of the ground unit is next checked (via a call to AIRGRND). (For a description of AIRGRND, also see Section 5.3.2.) If the air unit is on an air strike mission scheduled to occur this subinterval, the ordnance deliver is accomplished (via a call to ADW), along with the computation of ground casualties. The entire process is repeated for all air units for one subinterval, then the next of the four subintervals is processed.

Each air unit is checked, via a call to CK4XING, to see if its flight during this minute violated any control measures. If so, CK4XING issues an alert for each violated control measure. Each air unit is also checked to see if it has incurred losses resulting from ground-air fire. If so, an alert is issued indicating the losses. Personnel manning the ground-air defense fire weapon are distributed over their personnel vulnerability classes as a function of whether or not the weapon has been fired any time this minute (via a call to AIRCAS1).

Brief descriptions of the subroutines called by AIRMOV2 follow:

a. AIRCAS

AIRCAS is called by AIRMOV2 wherever a ground unit has detected an enemy air unit during a particular quarter-minute subinterval, is in an air defense state allowing him to fire at the air unit, and has not yet fired his air defense weapons this quarter-minute. AIRCAS computes the number of rounds fired by each air defense weapon in the ground unit and their effect. The general logic flow of AIRCAS is illustrated in Figure 5-127.

For each air defense weapon in the ground unit, each point of the air unit's movement during that quarter-minute (two endpoints plus any check points between them) is examined pairwise. If it passes three

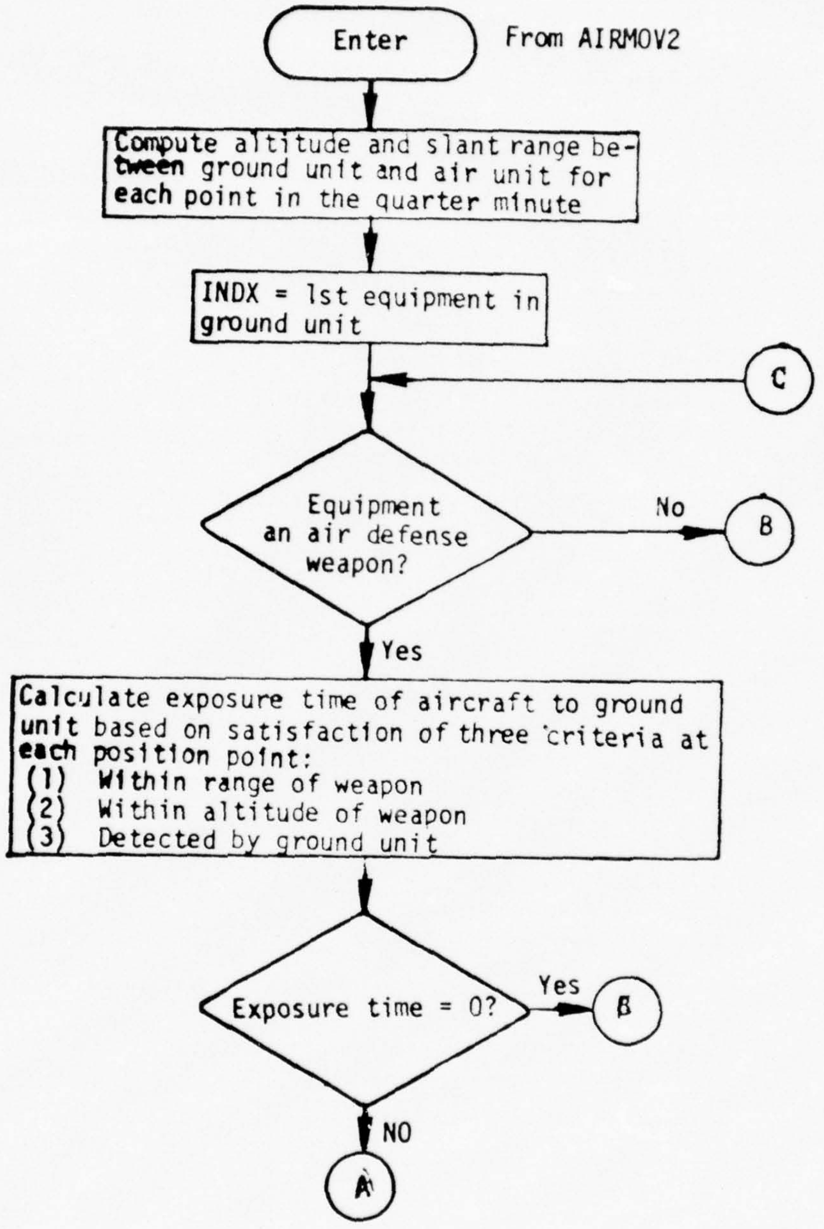

Enter
From AIRMOV2
Compute altitude and slant range between ground unit and air unit for each point in the quarter minute
INDX = 1st equipment in ground unit
C
Equipment an air defense weapon?
No
B
Yes
Calculate exposure time of aircraft to ground unit based on satisfaction of three criteria at each position point:
(1) Within range of weapon
(2) Within altitude of weapon
(3) Detected by ground unit
Exposure time = 0?
Yes
B
NO
A

Figure 5-127. General Logic Flow of Subroutine AIRCAS

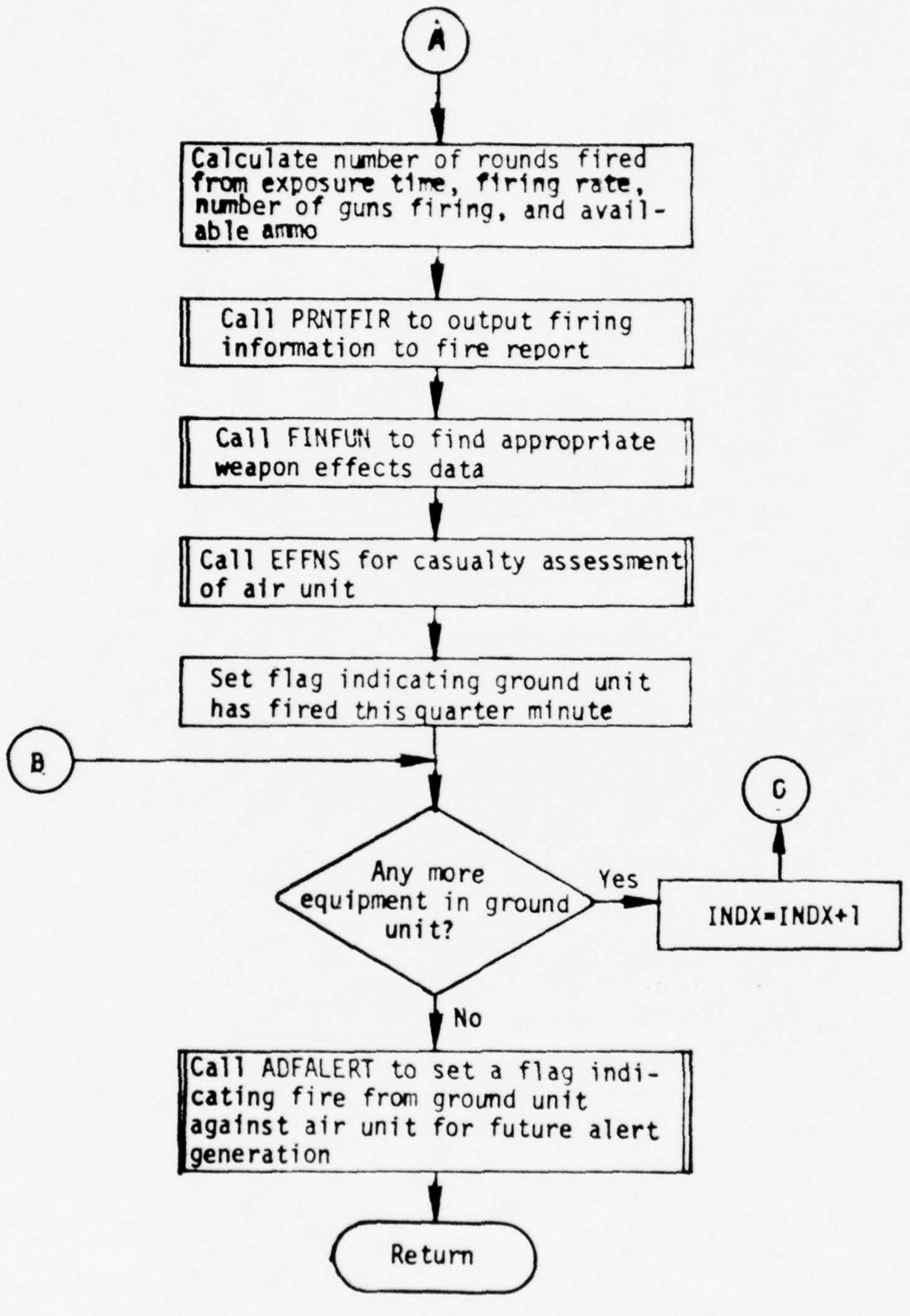

A
Calculate number of rounds fired from exposure time, firing rate, number of guns firing, and available ammo
Call PRNTFIR to output firing information to fire report
Call FINFUN to find appropriate weapon effects data
Call EFFNS for casualty assessment of air unit
Set flag indicating ground unit has fired this quarter minute
B
Any more equipment in ground unit?
Yes
INDX=INDX+1
C
No
Call ADFALERT to set a flag indicating fire from ground unit against air unit for future alert generation
Return

Figure 5-127. General Logic Flow of Subroutine AIRCAS (Continued)

checks; namely, 1) within altitude range of weapon, 2) within slant range of weapon, 3) detection occurred at the point, then the point is counted. If the other point of the pair is also counted, firing occurs for the whole interval of time. If just one point of a pair passes, the duration of firing over that pair of points is generated stochastically from a uniform distribution. All pairs of points in the quarter-minute are examined this way, and the total duration of fire is computed. This duration is multiplied by the firing rate per gun, the number of such guns in the unit, and the suppression factor of the unit to yield the total number of rounds fired. Ammunition levels are reduced for the weapon fired. PRNTFIR is called to save appropriate data for the air defense fire report, FINFUN is called to find the weapons effects function used to compute possible air losses, and EFFNS performs the actual casualty computation. ADFALERT is called to generate appropriate air defense alerts.

b. ADFALERT

This subroutine is called either by AIRMOV2 or AIRCAS for each ground unit/opposite colored air unit pairing. If the ground unit has changed either from a firing to non-firing or non-firing to firing state, an alert is issued. Otherwise, it simply returns. The general logic flow of ADFALERT is presented as Figure 5-128.

c. ADW

The philosophy governing air ordnance delivery has been modified since the CATTS system was originally delivered to the Army in May, 1975, per Army request. Originally air units delivered ordnance against a specified unit and all equipment in the unit, as well as personnel, was examined to assess the effect of the ordnance. No specific equipment type in the target unit was attacked directly. After using the system for approximately four months, the Army came to the conclusion that the delivery of air ordnance could only be modeled realistically if specific equipment types were attacked. This was especially true for guided weapons, which are by far the most effective. This change affected the logic of subroutine ADW. Therefore, the description that follows, and the reference figure containing the logic flow of the subroutine, differ considerably from the documentation in the CATTS Programming Report.

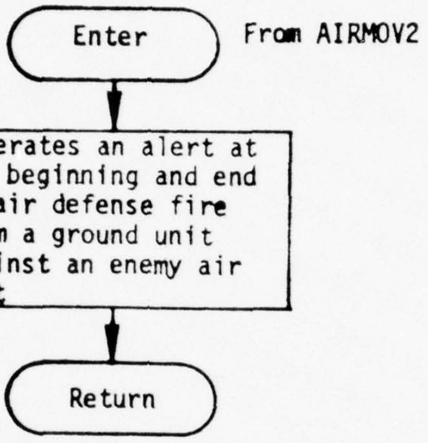

Enter
From AIRMOV2
Generates an alert at the beginning and end of air defense fire from a ground unit against an enemy air unit
Return

Figure 5-128. General Logic Flow of Subroutine ADFALERT

ADW is called by AIRMOV2 when the calling routine is processing an air unit having a designated ordnance deliver point in the quarter-minute subinterval currently being processed. The general logic flow of ADW is presented as Figure 5-129.

ADW uses ordnance attributes to compute the number of separate drops required to deliver the ordnance, the distance between drops if more than one is required, and the total amount of ordnance delivered per drop considering the number of aircraft in the air unit. The probability of a dud is accounted for by removing duds from the load based on a random number draw. Average ballistic errors are calculated (via a call to ADWDATA) in range and deflection relative to the air unit's direction of movement, and subjected to a normally distributed random number generator with zero mean. The range and deflection errors are transformed to the X, Y plane in order to compute the impact point, as depicted in Figure 5-130. An alert is issued indicating the impact point and type of ordnance. The impact point is stored for impacting fires display the next minute. A lethal area around the impact point is computed, as shown in Figure 5-131, for the ordnance from parameters computed by ADWDATA from mean area of effectiveness against personnel standing in the open. For air units having a non-unit target (bridge, road, or area target), each ground unit, both friendly and enemy, is checked for overlap with the lethal area (via a call to FRFD). Any unit having a non-zero overlap is flagged. Personnel casualties are computed by means of an ordnance-defined fraction of casualties in the overlap area for each vulnerability class. Casualties are integerized via a random number call, and appropriate casualty tables are updated. Equipment losses for equipments in class 1(IEQCLS = 1), soft targets, are computed similarly. Each equipment type has an attribute (IEQCLS) describing its vulnerability to an air strike. Only target equipments with this attribute set for soft target (=1) are processed. Soft targets have a fraction defined that is used to determine the lethal overlap area. Equipment losses are integerized and appropriate tables updated. If an equipment loss carries with it related personnel casualties, personnel are removed from the vulnerability class having the largest number of personnel. An alert is generated at the completion of the strike for each unit incurring casualties. After all units have been completed,

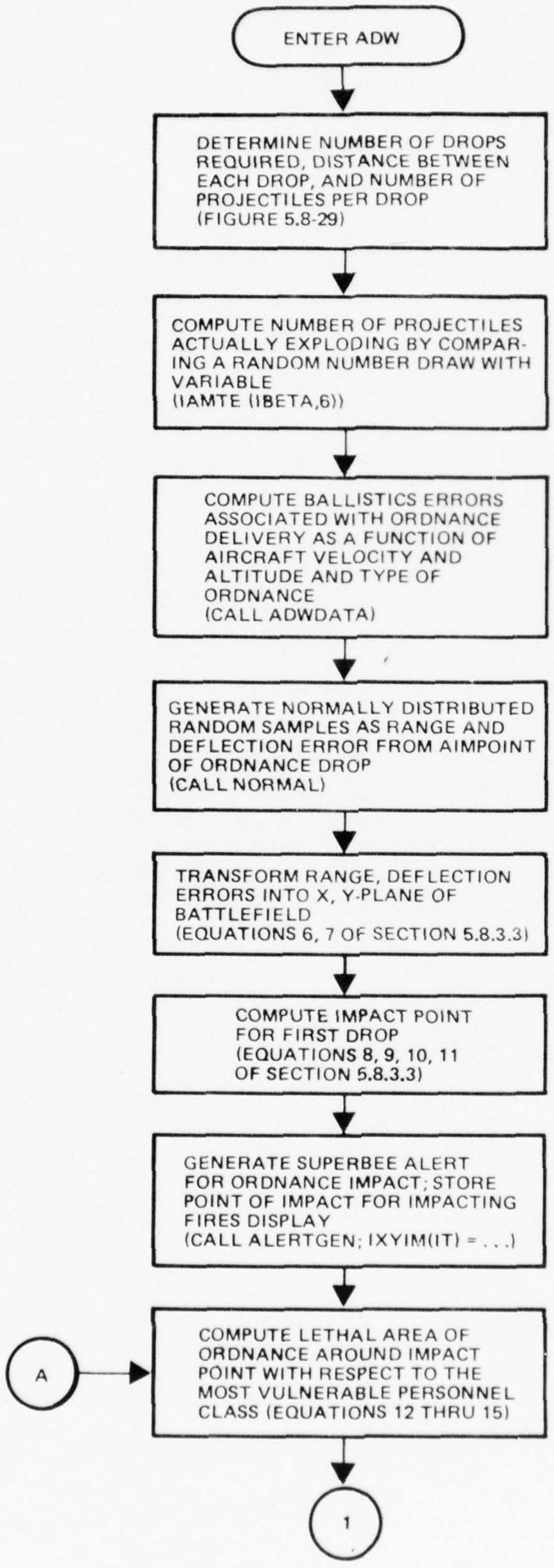

ENTER ADW
DETERMINE NUMBER OF DROPS REQUIRED, DISTANCE BETWEEN EACH DROP, AND NUMBER OF PROJECTILES PER DROP (FIGURE 5.8-29)
COMPUTE NUMBER OF PROJECTILES ACTUALLY EXPLODING BY COMPARING A RANDOM NUMBER DRAW WITH VARIABLE (IAMTE (IBETA,6))
COMPUTE BALLISTICS ERRORS ASSOCIATED WITH ORDNANCE DELIVERY AS A FUNCTION OF AIRCRAFT VELOCITY AND ALTITUDE AND TYPE OF ORDNANCE (CALL ADWDATA)
GENERATE NORMALLY DISTRIBUTED RANDOM SAMPLES AS RANGE AND DEFLECTION ERROR FROM AIMPOINT OF ORDNANCE DROP (CALL NORMAL)
TRANSFORM RANGE, DEFLECTION ERRORS INTO X, Y-PLANE OF BATTLEFIELD (EQUATIONS 6, 7 OF SECTION 5.8.3.3)
COMPUTE IMPACT POINT FOR FIRST DROP (EQUATIONS 8, 9, 10, 11 OF SECTION 5.8.3.3)
GENERATE SUPERBEE ALERT FOR ORDNANCE IMPACT; STORE POINT OF IMPACT FOR IMPACTING FIRES DISPLAY (CALL ALERTGEN; IXYIM(IT) = . . .)
A
COMPUTE LETHAL AREA OF ORDNANCE AROUND IMPACT POINT WITH RESPECT TO THE MOST VULNERABLE PERSONNEL CLASS (EQUATIONS 12 THRU 15)
1

Figure 5-129. General Logic Flow of Subroutine ADW

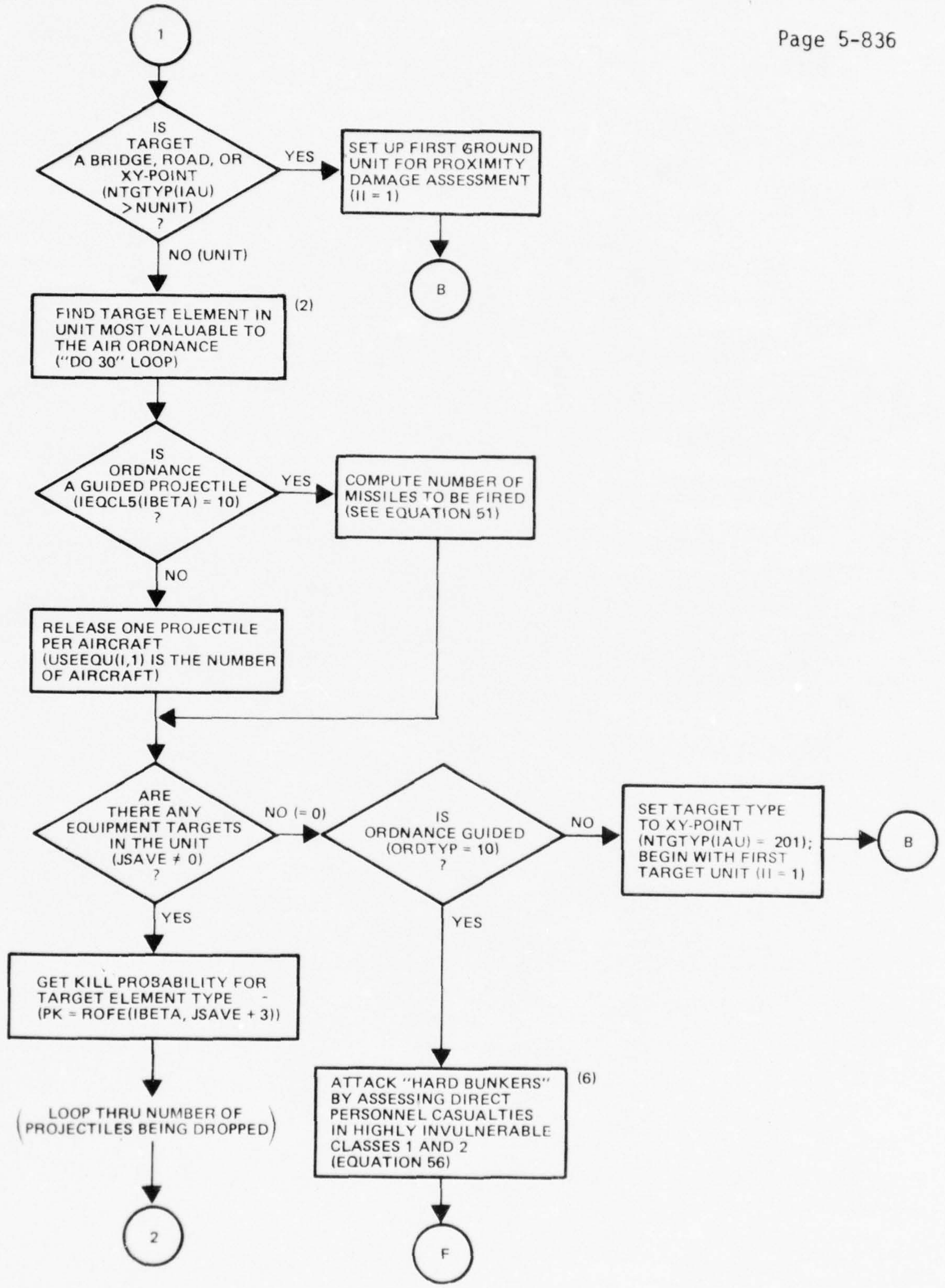

1
IS TARGET A BRIDGE, ROAD, OR XY-POINT (NTGTYP(IAU) > NUNIT) ?
YES
SET UP FIRST GROUND UNIT FOR PROXIMITY DAMAGE ASSESSMENT (II = 1)
B
NO (UNIT)
FIND TARGET ELEMENT IN UNIT MOST VALUABLE TO THE AIR ORDNANCE ("DO 30" LOOP)
(2)
IS ORDNANCE A GUIDED PROJECTILE (IEQCL5(IBETA) = 10) ?
YES
COMPUTE NUMBER OF MISSILES TO BE FIRED (SEE EQUATION 51)
NO
RELEASE ONE PROJECTILE PER AIRCRAFT (USEEQU(I,1) IS THE NUMBER OF AIRCRAFT)
ARE THERE ANY EQUIPMENT TARGETS IN THE UNIT (JSAVE ≠ 0) ?
NO (= 0)
IS ORDNANCE GUIDED (ORDTYP = 10) ?
NO
SET TARGET TYPE TO XY-POINT (NTGTYP(IAU) = 201); BEGIN WITH FIRST TARGET UNIT (II = 1)
B
YES
GET KILL PROBABILITY FOR TARGET ELEMENT TYPE (PK = ROFE(IBETA, JSAVE + 3))
YES
ATTACK "HARD BUNKERS" BY ASSESSING DIRECT PERSONNEL CASUALTIES IN HIGHLY INVULNERABLE CLASSES 1 AND 2 (EQUATION 56)
(6)
LOOP THRU NUMBER OF PROJECTILES BEING DROPPED
2
F

Figure 5-129. General Logic Flow of Subroutine ADW (Cont'd.)

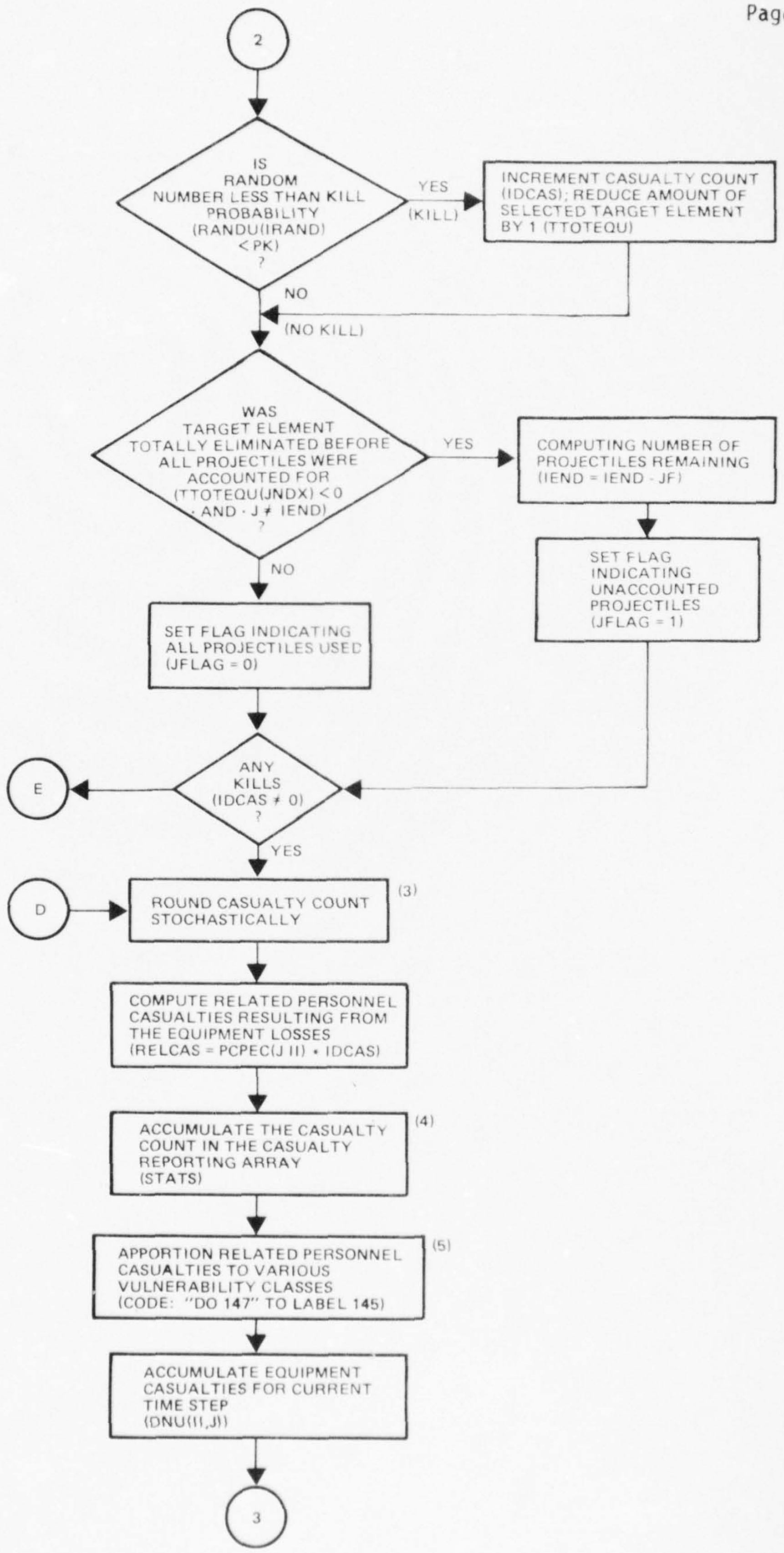

2
IS RANDOM NUMBER LESS THAN KILL PROBABILITY (RANDU(IRAND) < PK) ?
YES (KILL)
INCREMENT CASUALTY COUNT (IDCAS); REDUCE AMOUNT OF SELECTED TARGET ELEMENT BY 1 (TTOTEQU)
NO (NO KILL)
WAS TARGET ELEMENT TOTALLY ELIMINATED BEFORE ALL PROJECTILES WERE ACCOUNTED FOR (TTOTEQU(JNDX) < 0 .AND. J ≠ IEND) ?
YES
COMPUTING NUMBER OF PROJECTILES REMAINING (IEND = IEND - JF)
SET FLAG INDICATING UNACCOUNTED PROJECTILES (JFLAG = 1)
NO
SET FLAG INDICATING ALL PROJECTILES USED (JFLAG = 0)
ANY KILLS (IDCAS ≠ 0) ?
E
YES
D
ROUND CASUALTY COUNT STOCHASTICALLY
(3)
COMPUTE RELATED PERSONNEL CASUALTIES RESULTING FROM THE EQUIPMENT LOSSES (RELCAS = PCPEC(J II) * IDCAS)
ACCUMULATE THE CASUALTY COUNT IN THE CASUALTY REPORTING ARRAY (STATS)
(4)
APPORTION RELATED PERSONNEL CASUALTIES TO VARIOUS VULNERABILITY CLASSES (CODE: "DO 147" TO LABEL 145)
(5)
ACCUMULATE EQUIPMENT CASUALTIES FOR CURRENT TIME STEP (DNU(II,J))
3

Figure 5-129. General Logic Flow of Subroutine ADW (Cont'd.)

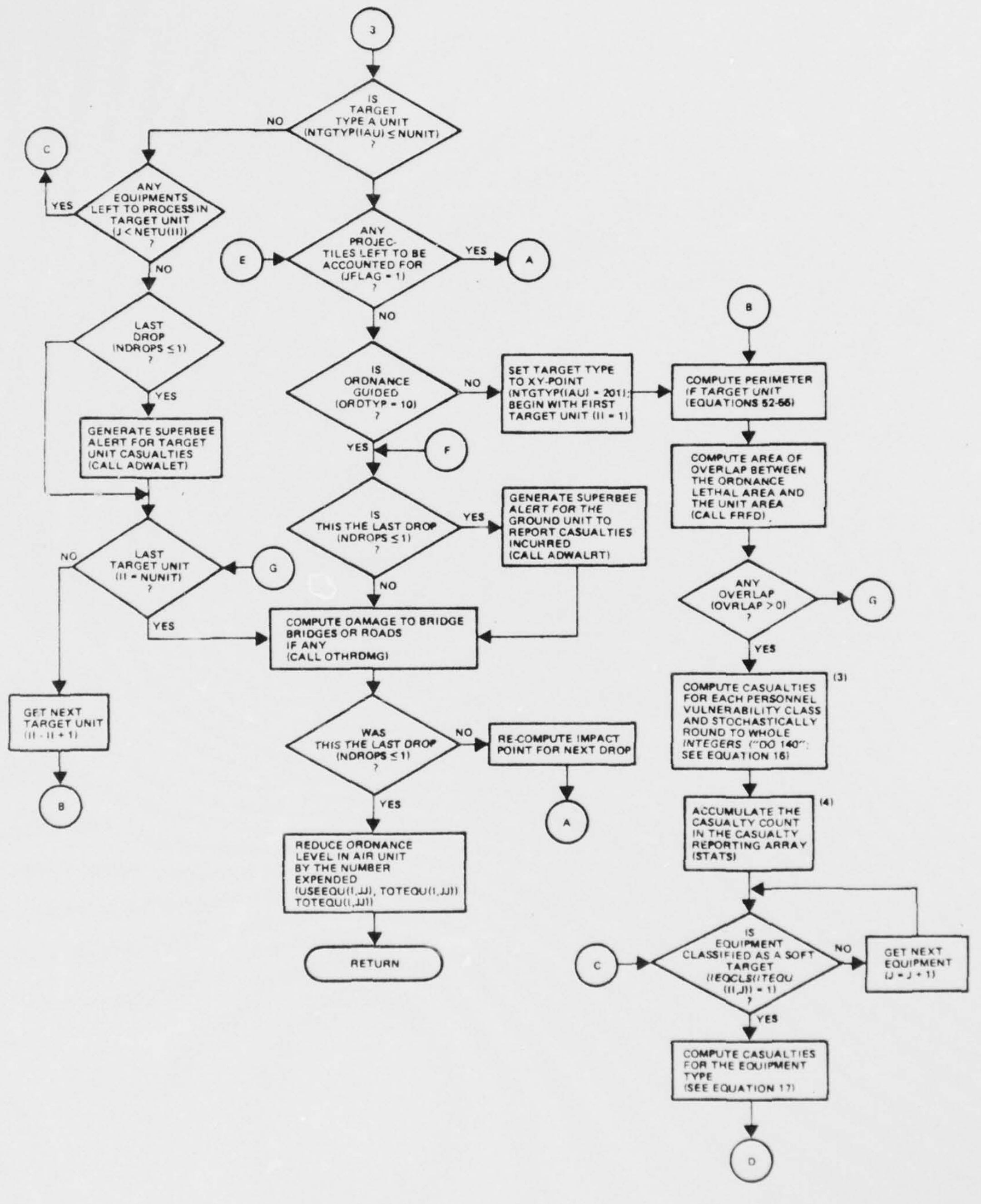

3
IS TARGET TYPE A UNIT (NTGTYP(IAU) ≤ NUNIT) ?
NO
C
YES
ANY EQUIPMENTS LEFT TO PROCESS IN TARGET UNIT (J < NETU(II)) ?
NO
LAST DROP (NDROPS ≤ 1) ?
YES
GENERATE SUPERBEE ALERT FOR TARGET UNIT CASUALTIES (CALL ADWALET)
LAST TARGET UNIT (II = NUNIT) ?
NO
YES
G
GET NEXT TARGET UNIT (II = II + 1)
B
E
ANY PROJECTILES LEFT TO BE ACCOUNTED FOR (JFLAG = 1) ?
YES
A
NO
IS ORDNANCE GUIDED (ORDTYP = 10) ?
NO
SET TARGET TYPE TO XY-POINT (NTGTYP(IAU) = 201); BEGIN WITH FIRST TARGET UNIT (II = 1)
YES
F
IS THIS THE LAST DROP (NDROPS ≤ 1) ?
YES
GENERATE SUPERBEE ALERT FOR THE GROUND UNIT TO REPORT CASUALTIES INCURRED (CALL ADWALRT)
NO
COMPUTE DAMAGE TO BRIDGE BRIDGES OR ROADS IF ANY (CALL OTHRDMG)
WAS THIS THE LAST DROP (NDROPS ≤ 1) ?
NO
RE-COMPUTE IMPACT POINT FOR NEXT DROP
A
YES
REDUCE ORDNANCE LEVEL IN AIR UNIT BY THE NUMBER EXPENDED (USEEQU(I,JJ), TOTEQU(I,JJ) TOTEQU(I,JJ))
RETURN
B
COMPUTE PERIMETER IF TARGET UNIT (EQUATIONS 52-56)
COMPUTE AREA OF OVERLAP BETWEEN THE ORDNANCE LETHAL AREA AND THE UNIT AREA (CALL FRFD)
ANY OVERLAP (OVRLAP > 0) ?
G
YES
(3)
COMPUTE CASUALTIES FOR EACH PERSONNEL VULNERABILITY CLASS AND STOCHASTICALLY ROUND TO WHOLE INTEGERS ("DO 140"; SEE EQUATION 16)
(4)
ACCUMULATE THE CASUALTY COUNT IN THE CASUALTY REPORTING ARRAY (STATS)
C
IS EQUIPMENT CLASSIFIED AS A SOFT TARGET (IEQCLS(ITEQU (II,JI)) = 1) ?
NO
GET NEXT EQUIPMENT (J = J + 1)
YES
COMPUTE CASUALTIES FOR THE EQUIPMENT TYPE (SEE EQUATION 17)
D

Figure 5-129. General Logic Flow of Subroutine ADW (Cont'd.)

Notes to Figure 5-129:

(1) The number of total projectiles of the type to be fired from the air unit (USEEQU(I,JJ)) is compared with the quantity resulting from the number of projectiles usually dropped in a standard delivery per aircraft (IAMTE(IBETA,2)) times the number of aircraft (USEEQU(I,1)). If the first quantity (USEEQU(I,JJ)) is smaller, only one drop is required. Otherwise, the number of drops is taken from variable IAMTE(IBETA,4), the distance between drops from IAMTE(IBETA,5), the total number of projectiles over all drops (AMTFIRED) is computed as the number usually dropped in a standard delivery per aircraft times the number of aircraft, and the number delivered per drop (IEND) as the number of projectiles per drop per aircraft times the number of aircraft.

(2) The target equipment with the highest classification as an air target (IEQCLS) for which the projectile has a non-zero kill probability is used as the selected target element. If more than one type of target element has the highest classification, the one with the largest weighting factor (UBE) is chosen.

(3) If the random number draw, RANDU(IRAND), is less than the fractional part of the casualty accumulations, increment the integer part by 1; otherwise, truncate the fractional part.

(4) See Section 4 for a description of the STATS array.

(5) Personnel are removed from the vulnerability class having the largest number first, then the next largest number, and so on until all casualties are accounted for.

(6) If no bonafide equipment targets remain in a target unit and some guided projectiles remain unused, the remaining ones are used against "hard bunkers". Hard bunkers as such are not modeled as entities in CATTS. However, to satisfy a model enhancement requested by U. S. Army and U. S. Air Force personnel in Fort Benning, Georgia, this feature was added. The variable ROFE(IBETA,1) was defined to represent the number of personnel casualties resulting from a guided projectile destroying a hard bunker or personnel fortification. To each unused guided projectile is attributed the number of casualties specified by ROFE(IBETA,1) up to a maximum equal to the sum of the number of personnel in vulnerability classes 1 and 2 (see equation 56 in Section 5.8.3.3).

Figure 5-129. General Logic Flow of Subroutine ADW (Cont'd.)

$\rightarrow$ = aircraft flight path

A = aim point

I = impact point

R = range error

D = deflection error

$(R, D) = N(F_1(V,Z,T), F_2(V,Z,T))$

F_1, F_2 = table look up functions

N = stochastic normal distribution function

V = aircraft speed

Z = aircraft altitude

T = ordnance type

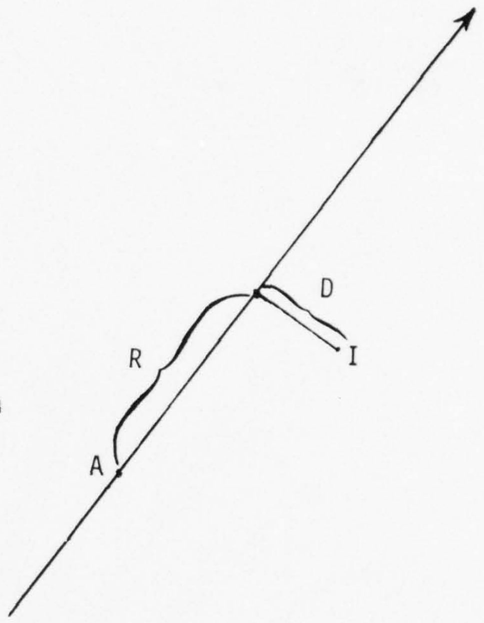
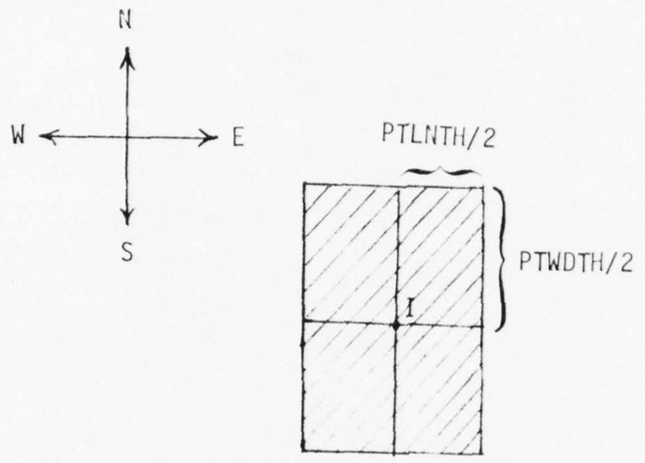


Figure 5-130. Calculation of Impact Point of Air Delivered Ordnance



I = calculated impact point

PTLNTH = $\sqrt{A/3}$

PTWDTH = 3 * PTLNTH

A = calculated "Lethal area" of ordnance

= F(V,Z,T)

V = aircraft speed

Z = aircraft altitude

T = ordnance type

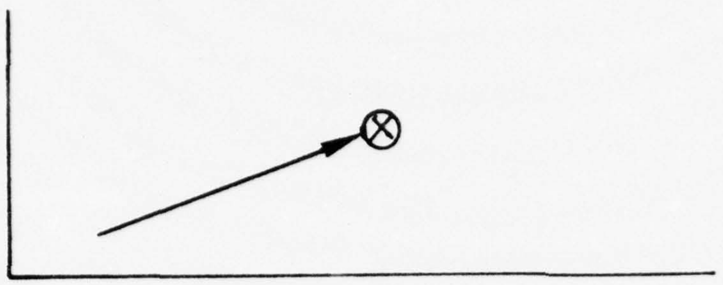
Figure 5-131. Calculation of Impact Area of Air Delivered Weapons

bridges and roads are checked to determine if the impact point of each drop affects any bridge or road defined in the data base (via a call to OTHRDMG). If any bridge or road segment is affected, damage is computed (via a call to DIDITHIT). If the strike is not yet completed, a separation distance "delta" is added to the last impact point, the next point of impact is calculated, and the casualty assessment process is repeated.

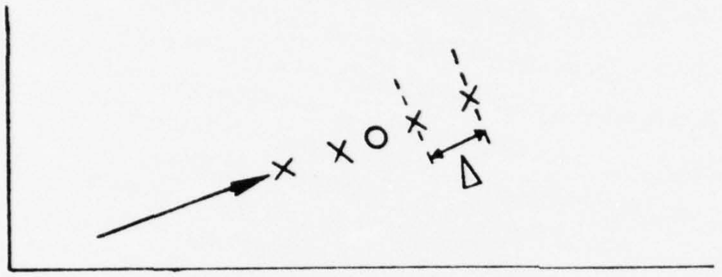
For air units having a unit, or hard, target designated, the equipment type in the target unit with the highest equipment class (representing its value as an air target) is selected for direct attack. If more than one equipment type has the highest classification, the equipment weighting factor (UBE) is used to determine the equipment type. The kill probability is compared against a random number to determine whether or not that particular projectile was effective. All projectiles of a given drop are evaluated similarly until exhausted. If the equipment type is completely destroyed before all projectiles are used, the next qualifying target equipment is selected. In the special case where not all guided projectiles (ORDTYP = 10) have been used, but no equipment types qualify as hard targets, highly invulnerable personnel in vulnerability classes 1 and 2 (assumed to be in hard fortifications, such as bunkers) are used as direct targets. Note (6) to the subroutine ADW flow chart discusses this case in more detail. All casualties, both equipment and personnel, are accumulated in the casualty reporting array (STATS). For equipment casualties, related personnel casualties are also computed, representing the number of personnel killed when an equipment type is destroyed (for example, when a tank is destroyed, the crew operating the tank also incurs casualties).

When all direct target casualties have been completed, the loop described above for bridge, road, or area targets is entered to evaluate secondary (proximity) casualties resulting from the direct strike. All units are processed for possible secondary casualties. When this loop is completed, the casualty assessment due to an air unit delivering its ordnance is completed. The amount of ordnance delivered is removed from the air unit's ordnance load.

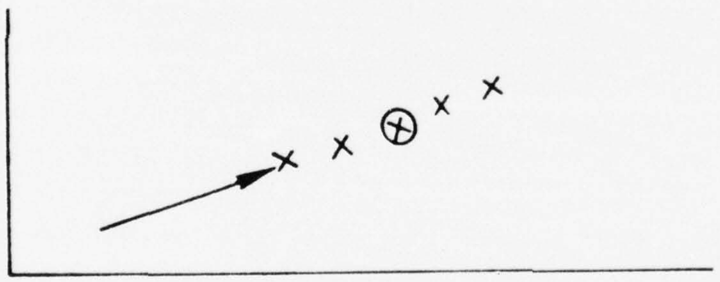
Figure 5-132 illustrates the strike pattern. Table 5-59 summarizes the lethality calculations made for air-delivered weapons.



a. Single Drop



b. Even Number of Drops



c. Odd Number of Drops

Key:

↗ –	Direction of flight
× –	Actual impact point
○ –	Calculated impact point
Δ –	Separation distance "delta"

Figure 5-132. Actual Impact Points for Three Air Strike Situations

Table 5-59. Casualty Calculations Made for Air Delivered Weapons

ORDNANCE	LETHAL AREA	CASUALTY COMPUTATION (FRACTION OF PEOPLE KILLED)

GP BOMBS	PL = $\sqrt{\text{Mean Area of Effectiveness/3}}$
	PW = 3 * Pattern Length
		1 - (1 - Overlap Area) ** (number of weapons)

CBU	PL = 2 * Weapon Radius + Stick Length + 480
	PW = Stick Width + 2 * Weapon Radius
		Overlap Area * $(1-e^{-(\text{Mean Area of Effectiveness}}$
		Weapon Radius))

ROCKETS/MACHINE GUN
(Fixed Wing Delivery)
(Rotary Wing Delivery)
	PL = X * Launch Range/sin(dive angle)
	PW = X * Launch Range
		Number of Rounds * Vulnerable Area
		* Overlap Area/ Number of Personnel

NAPALM	PL = $\sqrt{\text{2.2 * Mean Area of Effectiveness}}$
	PW = 0.21 * Pattern Length
		1 - (1 - Overlap Area) ** (number of weapons
		dropped)

GUIDED MISSILES	PL = $\sqrt{\text{Mean Area of Effectiveness}}$
	PW = Pattern Length
		Overlap Area

GUIDED GP BOMBS	Same as GP Bombs	Same as GP Bombs

d. ADWDATA

ADWDATA is called by ADW each time an air strike occurs. ADWDATA uses the air unit location at the time of ordnance delivery (including altitude and velocity) and the ordnance type being delivered to compute range and deflection probable errors and pattern length and width of lethal area. The general logic flow of ADWDATA is presented as Figure 5-133.

Air ordnance for the purposes of ballistic error lethal area calculations can be grouped in the following categories:

(1) low-drag bombs and napalm

(2) hi-drag bombs

(3) guided missiles

(4) CBU-2

(5) Rockeye and CBU-24

(6) Rockets and Cannons

For low-drag bombs and napalm, the time of fall and impact angle are computed from the aircraft altitude and velocity. The impact angle is used along with the bomb coefficients to compute lethal area. Ballistic errors are a function of aircraft altitude and time of fall, along with the bomb coefficients. Hi-drag bombs are done similarly except that the time of fall, ground range between target and aircraft at weapon release, and impact angle are selected by table look-up for the appropriate velocity and altitude.

Guided missiles have zero ballistic errors. Their lethal area is constant such that the width = 3 x length.

For the CBU-2 cluster bomb, time of fall and slant range are computed as a function of altitude, which in turn are used along with the ordnance coefficients to determine ballistic errors. Lethal area of a CBU-2 is computed from a pre-determined constant, but modified as a function of the aircraft altitude at delivery. Lethal area width = 3 x length.

For the CBU-24 and Rockeye, the ballistic errors are identical and constant, whereas the CBU-24 has constant lethal area and a Rockeye's lethal area depends on the impact angle, which is determined by aircraft altitude and velocity at release. The lethal area width = 3 x length.

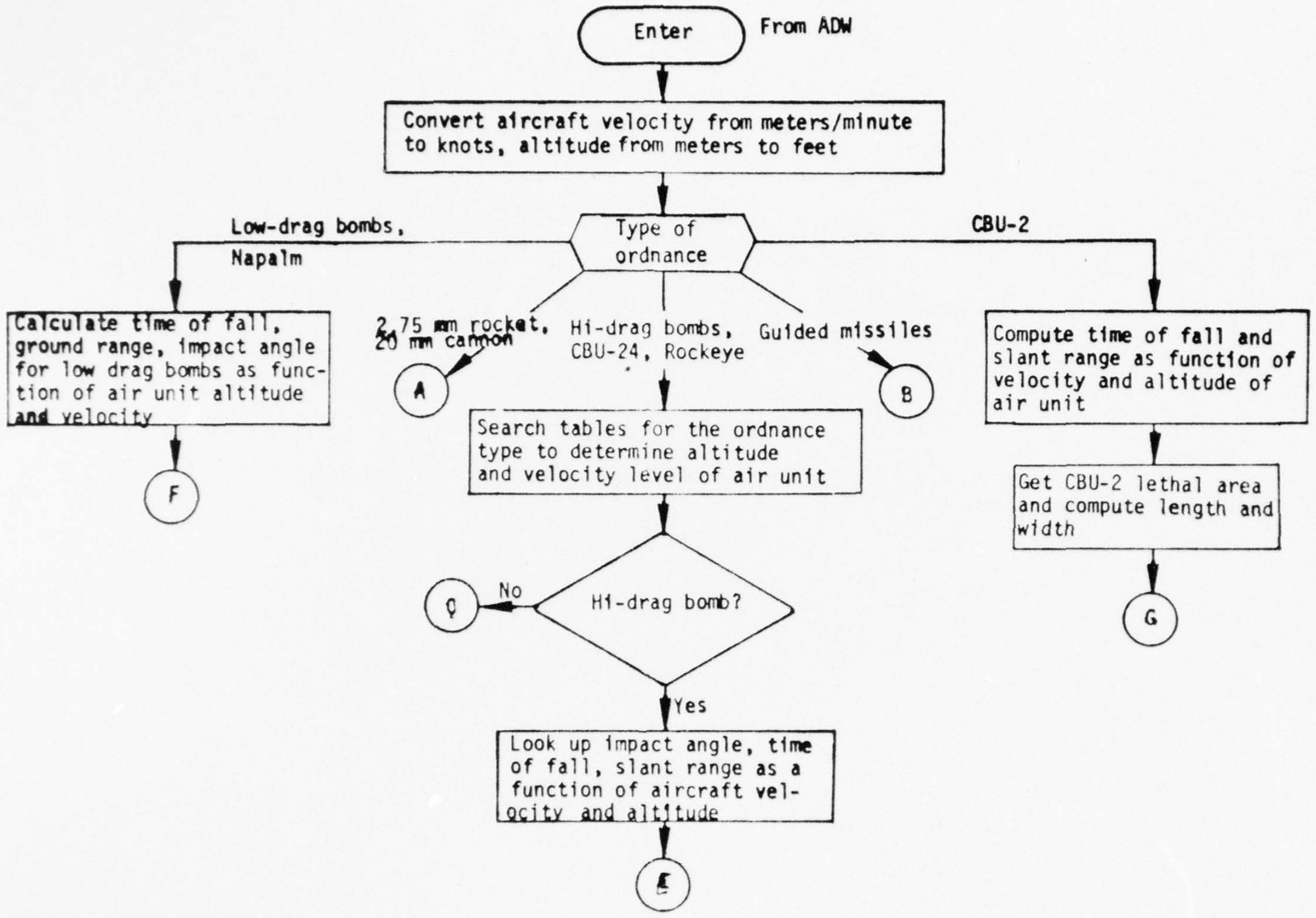

Enter
From ADW
Convert aircraft velocity from meters/minute to knots, altitude from meters to feet
Type of ordnance
Low-drag bombs, Napalm
CBU-2
2.75 mm rocket, 20 mm cannon
Hi-drag bombs, CBU-24, Rockeye
Guided missiles
Calculate time of fall, ground range, impact angle for low drag bombs as function of air unit altitude and velocity
F
A
B
Compute time of fall and slant range as function of velocity and altitude of air unit
Get CBU-2 lethal area and compute length and width
G
Search tables for the ordnance type to determine altitude and velocity level of air unit
Hi-drag bomb?
No
C
Yes
Look up impact angle, time of fall, slant range as a function of aircraft velocity and altitude
E

Figure 5-133. General Logic Flow of Subroutine ADWDATA

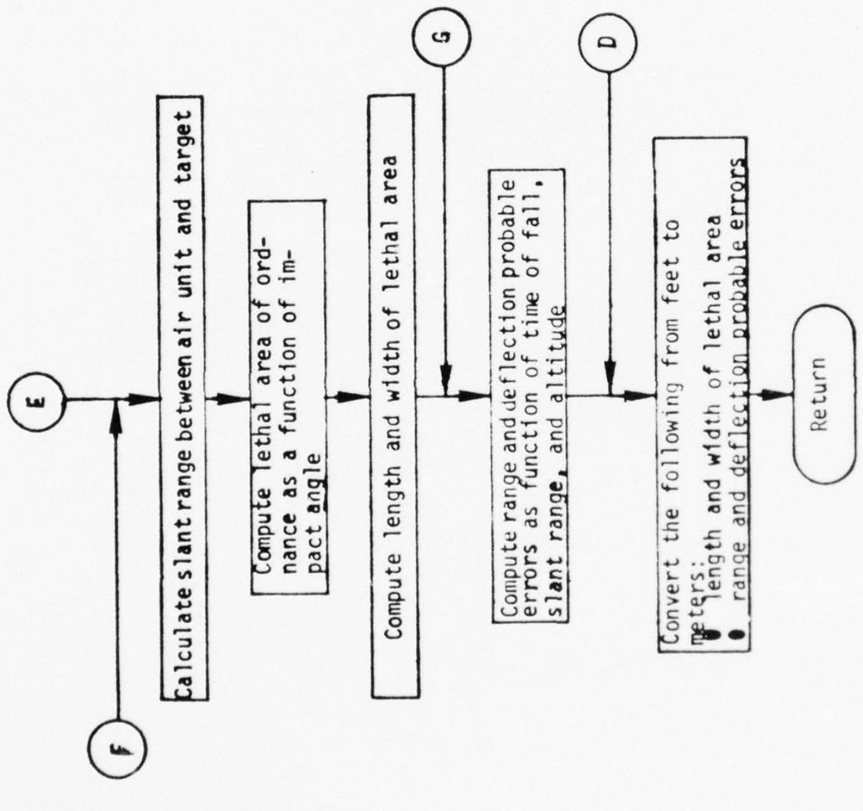

E
F
Calculate slant range between air unit and target
Compute lethal area of ordnance as a function of impact angle
Compute length and width of lethal area
G
Compute range and deflection probable errors as function of time of fall, slant range, and altitude
D
Convert the following from feet to meters:
● length and width of lethal area
● range and deflection probable errors
Return

Figure 5-133. General Logic Flow of Subroutine ADWDATA (Continued)

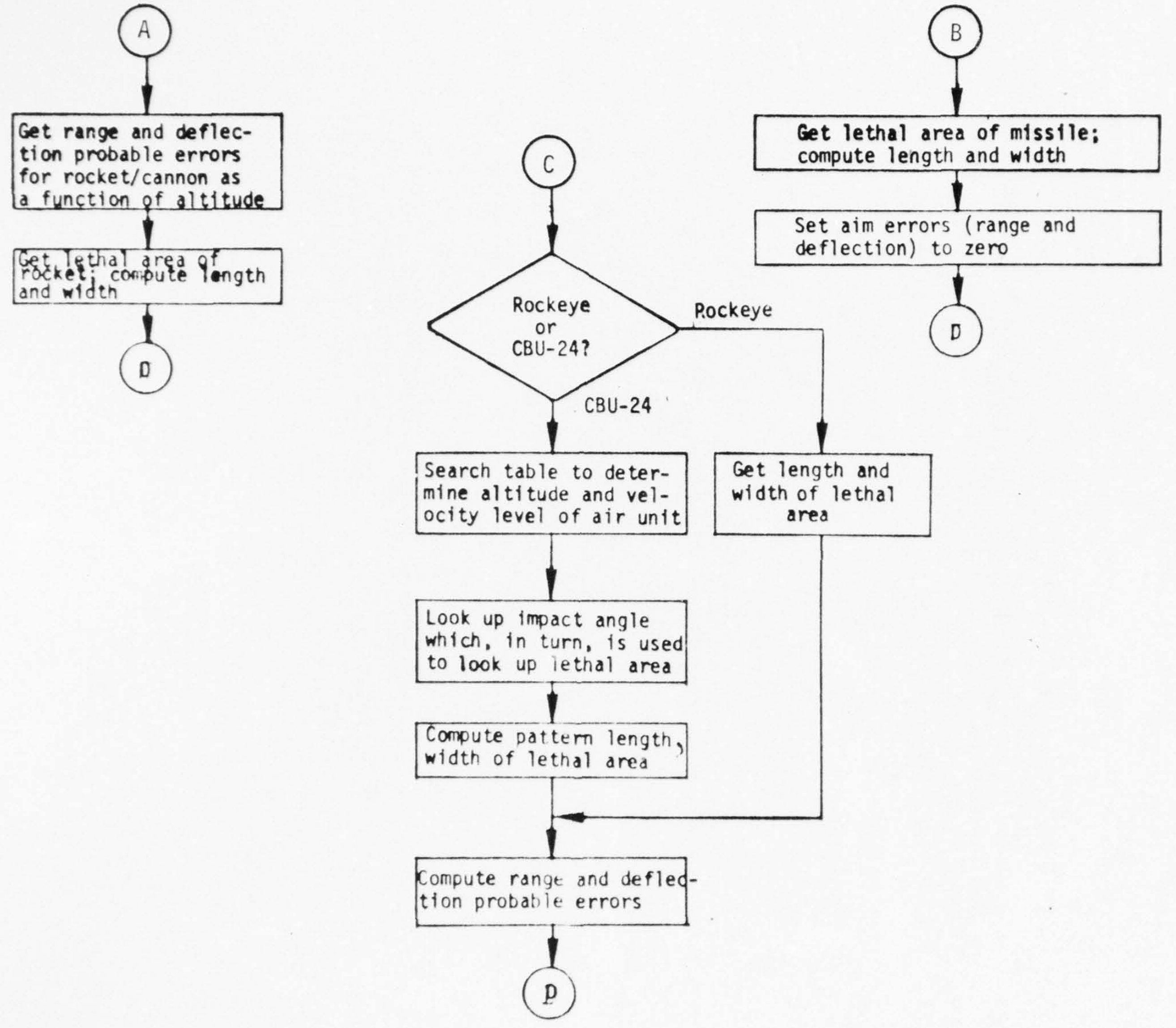

A
Get range and deflection probable errors for rocket/cannon as a function of altitude
Get lethal area of rocket; compute length and width
D
C
Rockeye or CBU-24?
Rockeye
CBU-24
Search table to determine altitude and velocity level of air unit
Look up impact angle which, in turn, is used to look up lethal area
Compute pattern length, width of lethal area
Get length and width of lethal area
Compute range and deflection probable errors
D
B
Get lethal area of missile; compute length and width
Set aim errors (range and deflection) to zero
D

Figure 5-133. General Logic Flow of Subroutine ADWDATA (Continued)

For the rockets and cannon, the lethal areas are constant but not identical. Ballistic errors are calculated as a function of altitude.

The equations for calculating lethal area for the different air-delivered ordnance types are presented in Table 5-59. A brief description of the data tables used in ADWDATA are presented in Table 5-60.

At the conclusion of processing, all units are converted back to meters for use by ADW.

e. NORMAL

This subroutine is given a mean and standard deviation as arguments, and produces a normally distributed random sample as a result. It accomplishes this by sampling 12 times from a uniform (0,1) random number generator, then performing the following computation:

RVNORM = (A - 6.0) * XSTDEV + XMEAN

where

RVNORM is the desired result

A is the cumulative sum of the 12 uniform (0,1) samples

XSTDEV is the standard deviation

XMEAN is the mean

The general logic flow of NORMAL is presented as Figure 5-134.

f. FRFD

Two parallel rectangles transformed to the X, Y coordinate system, one representing the unit, the other the lethal area, are checked to determine the area of overlap. This area is then divided by the unit area to yield the fraction of the unit in the lethal area. The calculation is quite simple. The maximum of the smaller X values of the two rectangles is subtracted from the minimum of the larger X values; similarly for Y. If either of these subtractions produces a non-positive result, there is zero overlap. Otherwise, the X and Y differences represent the sides of the overlap area, which is divided by the unit area to yield the desired result.

The general logic flow of FRFD is presented in Figure 5-135.

Table 5-60. ADWDATA Data Tables

Data Variable	Dimension	Description
IV	6	Table of "break" velocities (in knots).
IA	7	Table of "break" altitudes (in feet).
GR	(7, 6, 3)	Table of ground ranges (in feet) as a function of altitude, velocity and ordnance type.
TF	(7, 6, 3)	Table of time of fall as a function of altitude, velocity and ordnance type.
XMAEGP	(5, 2)	Table containing a maximum and minimum mean area of effectiveness as a function of impact angle for the five types of general purpose low-drag bombs.
LARKI	(2, 7)	Table of impact angles (1, IA) and lethal areas (2, LA) for seven break values of impact angle for the Rockeye.
IMA	(7, 6, 4)	Table of impact angles expressed as a function of altitude, velocity, and ordnance type (the three high-drag general purpose bombs and Rockeye).
CBU24RSD	(7, 6)	Table of standard deviations of impact error distances in range for the CBU24, given as a function of altitude and velocity.
CBU24DSD	(7, 6)	Table of standard deviations of impact error distance in deflection for the CBU24, given as a function of altitude and velocity.

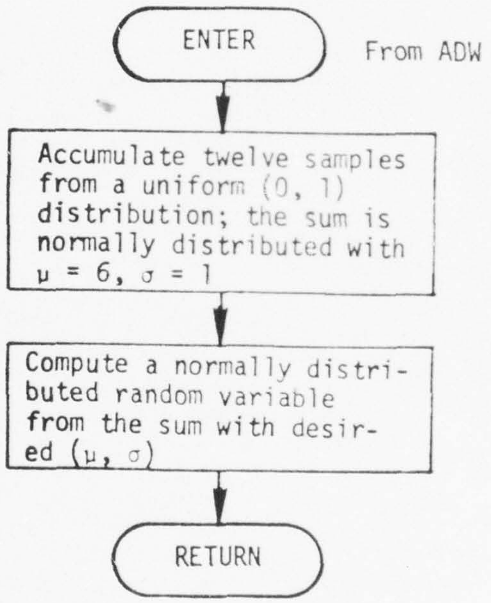
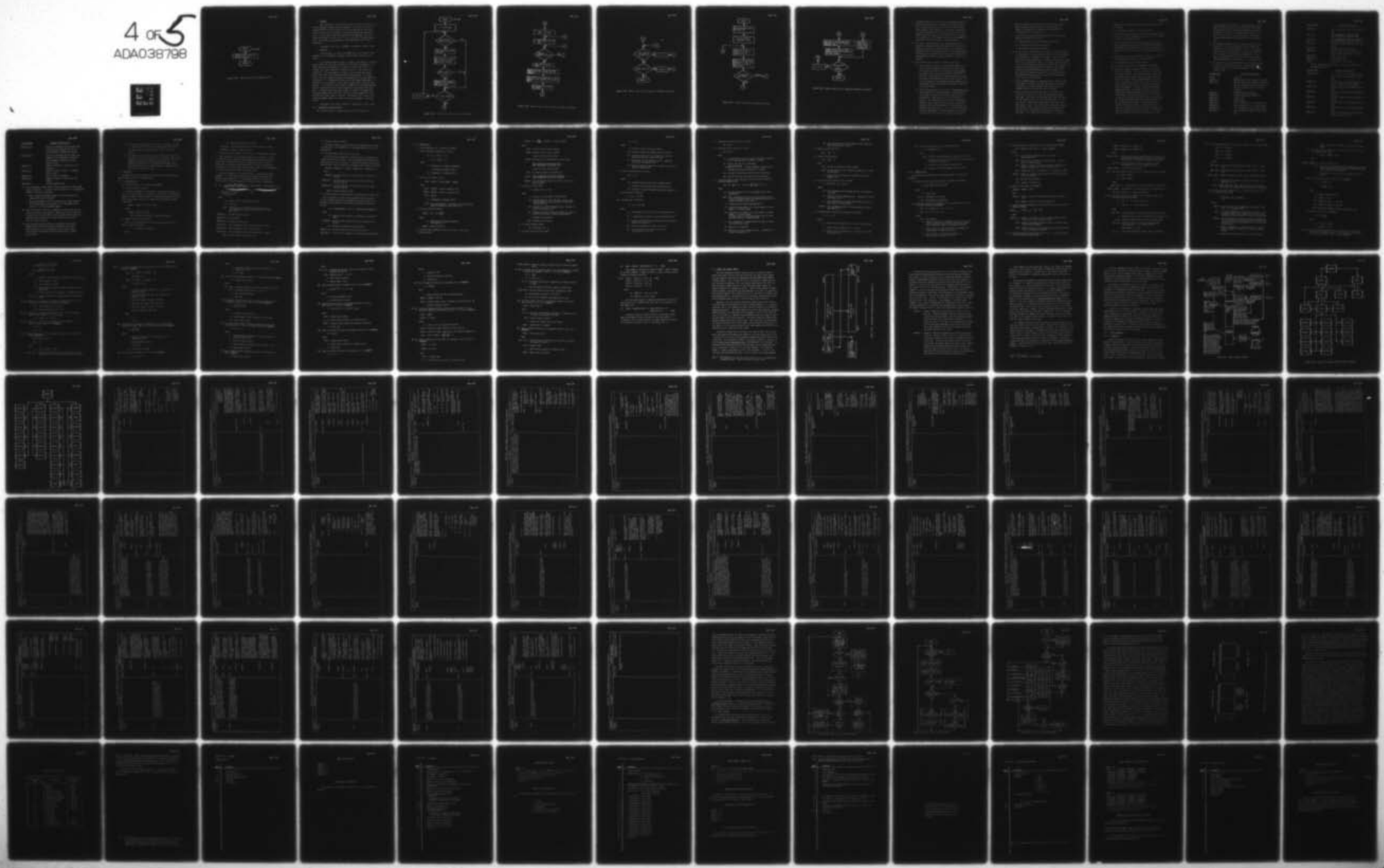

ENTER
From ADW
Accumulate twelve samples from a uniform (0, 1) distribution; the sum is normally distributed with μ = 6, σ = 1
Compute a normally distributed random variable from the sum with desired (μ, σ)
RETURN

Figure 5-134. General Logic Flow of Subroutine NORMAL

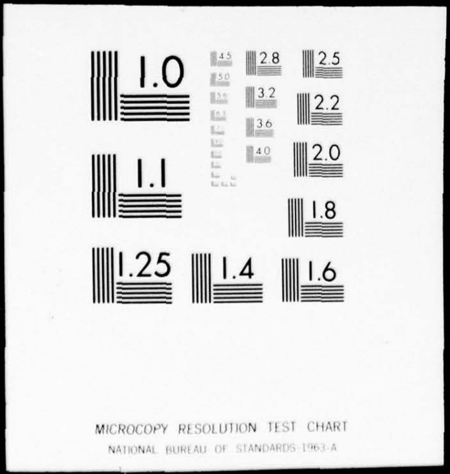
AD-A038 798 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

4 OF 5
ADA038798



4 OF 5

ADA038798



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

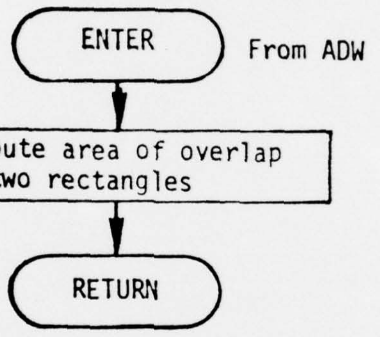


Figure 5-135. General Logic Flow of Subroutine FRFD

g. OTHRDMG

This subroutine is called for each drop of an air strike to assess possible damage to bridges and road segments. It sets up a call to subroutine DIDITHIT for each bridge not yet destroyed, and for any bridge affected by the drop (in DIDITHIT), accumulates damage and generates an alert. It then checks each road segment not yet destroyed. Any road within 1000 meters in both X and Y is sent to DIDITHIT for damage evaluation. The results are accumulated for the road segment and an alert is generated.

The general logic flow of OTHRDMG is presented as Figure 5-136.

h. DIDITHIT

This subroutine is called by OTHRDMG each time a bridge or road segment is to be evaluated for damage resulting from a drop of an air strike.

For evaluation against either a bridge or road segment, individual impact points for each piece of ordnance in the drop are computed. The impact locations are separated by 10 meters in both X and Y and, for bridges, each impact location is checked to see if it falls within the bridge area. For road segments, a crater is built around each impact point, the crater size being determined by the road type. The largest width of the road segment destroyed is accumulated for the road segment to a maximum of 1. In computing the fraction of road segment destroyed, it is necessary to first determine if the impact location fell inside the road segment. If so, half the crater radius is added to the difference between half the road width and the distance between the road center and impact point. If impact was off the road segment, it suffices to compute the difference between the crater radius and the distance between the impact point and the closest edge of the road. For bridges only, a flag is set indicating damage; whereas for roads, the fraction damaged is calculated as well.

The general logic flow of DIDITHIT is presented as Figure 5-137.

5.8.2 Assumptions and Data Sources

The assumptions used in constructing the CATTS Air Module are:

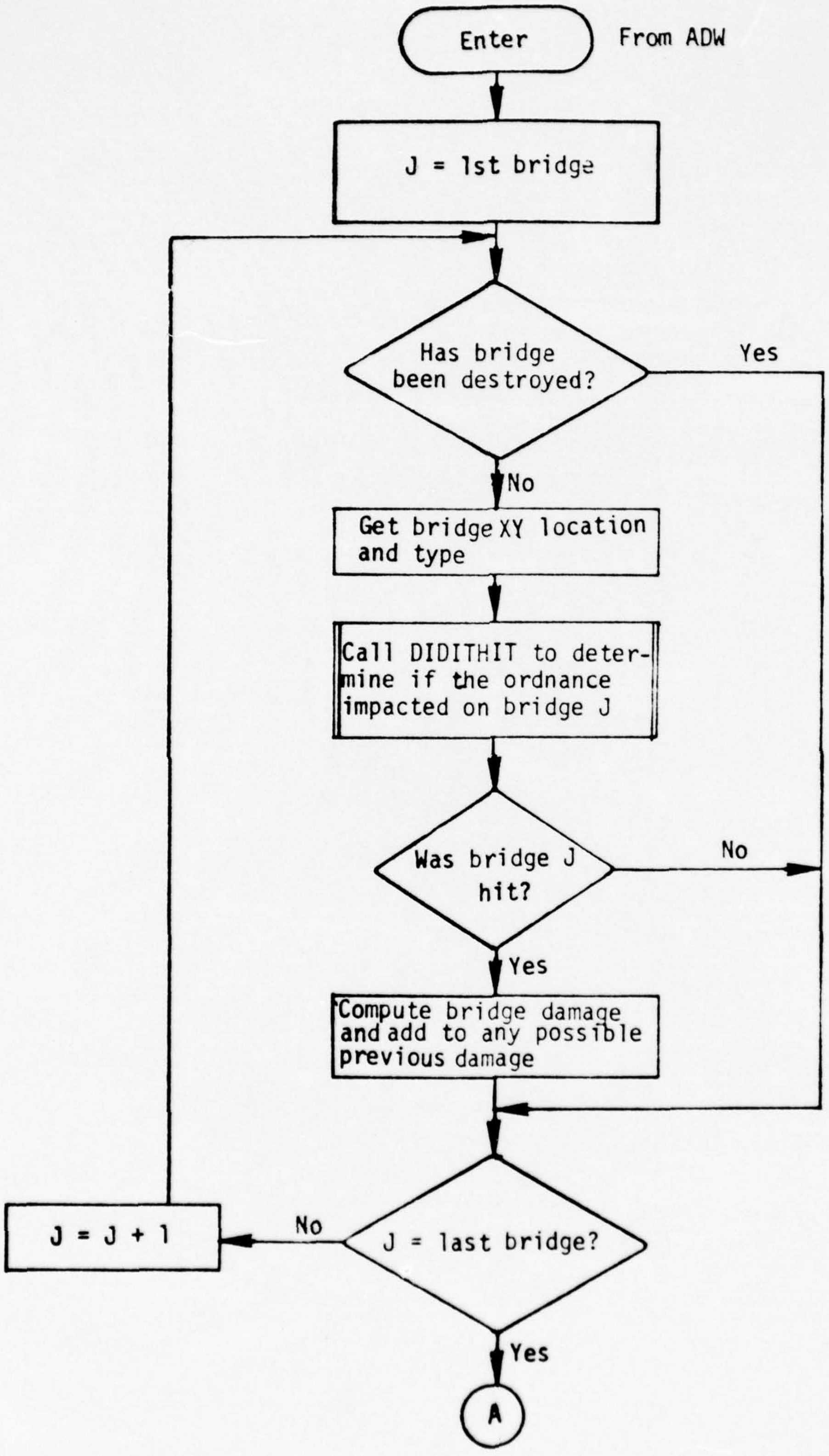
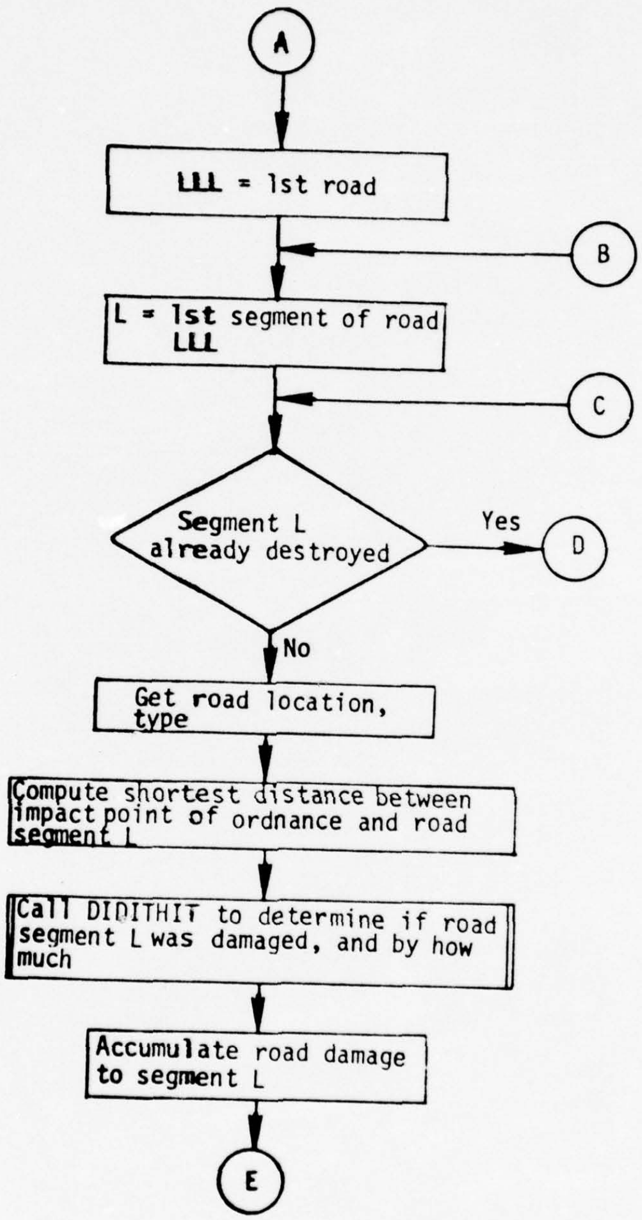

Enter
From ADW
J = 1st bridge
Has bridge been destroyed?
Yes
No
Get bridge XY location and type
Call DIDITHIT to determine if the ordnance impacted on bridge J
Was bridge J hit?
No
Yes
Compute bridge damage and add to any possible previous damage
J = last bridge?
No
J = J + 1
Yes
A

Figure 5-136. General Logic Flow of Subroutine OTHRDMG



A
LLL = 1st road
B
L = 1st segment of road LLL
C
Segment L already destroyed
Yes
D
No
Get road location, type
Compute shortest distance between impact point of ordnance and road segment L
Call DIDITHIT to determine if road segment L was damaged, and by how much
Accumulate road damage to segment L
E

Figure 5-136. General Logic Flow of Subroutine OTHRDMG (Continued)

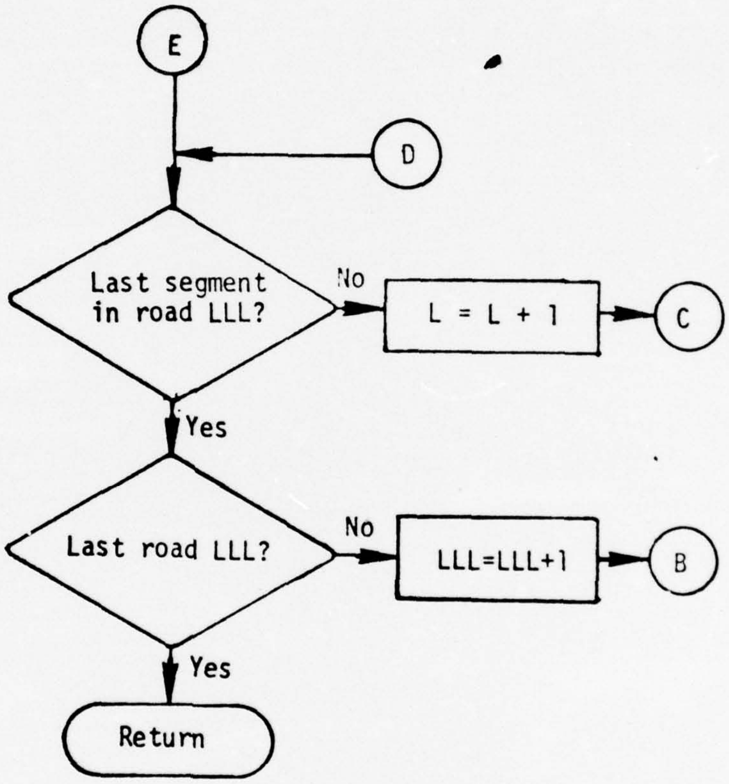


Figure 5-136. General Logic Flow of Subroutine OTHRDMG (Continued)

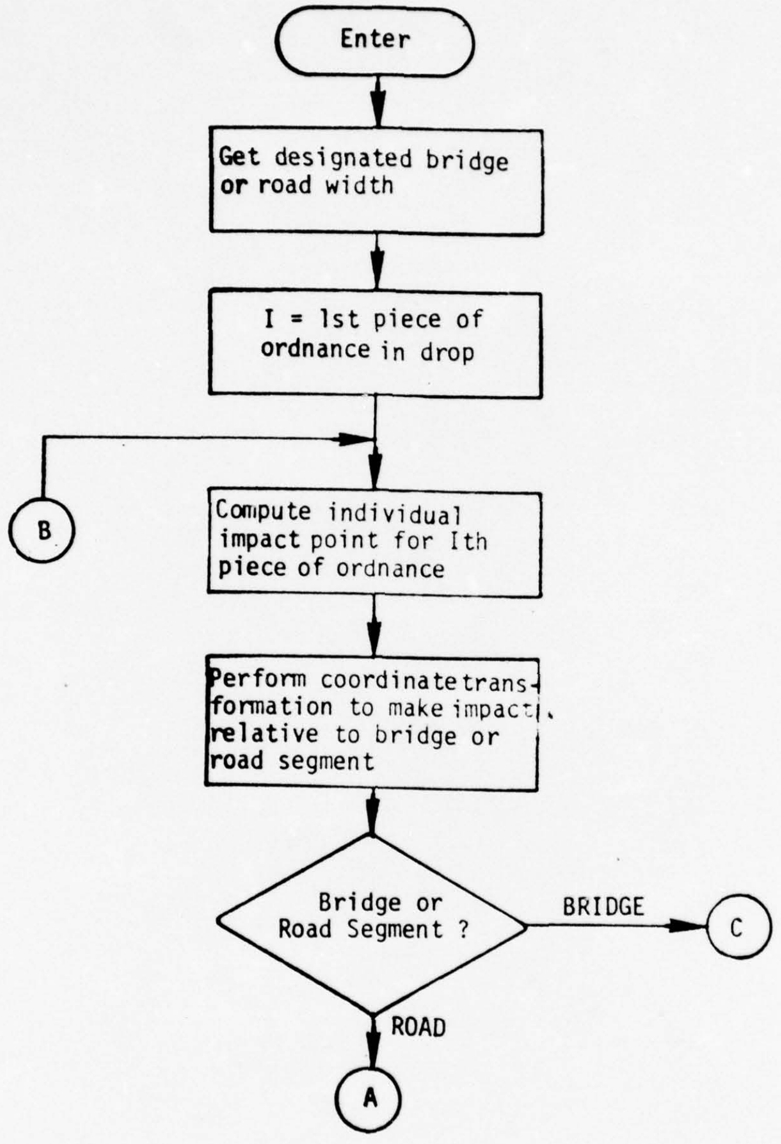

Enter
Get designated bridge or road width
I = 1st piece of ordnance in drop
Compute individual impact point for Ith piece of ordnance
Perform coordinate transformation to make impact relative to bridge or road segment
Bridge or Road Segment ?
BRIDGE
ROAD
A
B
C

Figure 5-137. General Logic Flow of Subroutine DIDITHIT

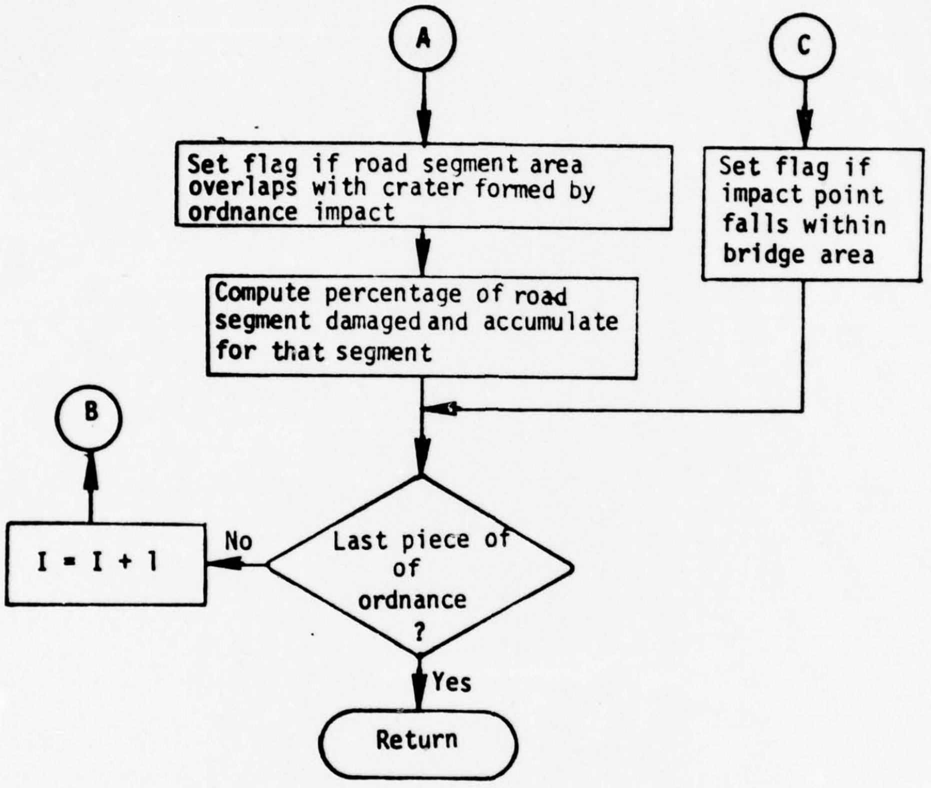

A
C
Set flag if road segment area overlaps with crater formed by ordnance impact
Set flag if impact point falls within bridge area
Compute percentage of road segment damaged and accumulate for that segment
B
I = I + 1
No
Last piece of of ordnance ?
Yes
Return

Figure 5-137. General Logic Flow of Subroutine DIDITHIT (Continued)

1) A maximum of ten air missions may be conducted concurrently. Each air mission becomes a unit while in progress, therefore, using up the space allotted for one of the 99 total units allowed in the simulation. The number of concurrent air missions allowed in any scenario is user input to any number less than or equal to the maximum of ten. Whatever this number may be, that number of units must be unused and reserved for use by air units as they are created.

2) Air units, and air missions, are created dynamically in response to command and control inputs from the controllers. Each air unit/mission exists only until it is completed. When the mission is completed, the air unit becomes defunct, and the space in the data tables used by the now-defunct air unit becomes available for use by the next air unit to be created.

3) Air missions can be "canned" in either of two ways.

   a) CATTS allows the input of pre-determined command and control events scheduled to occur at a specific time. In this way, a series of air strikes can be preplanned to happen at exactly the same time each time CATTS is exercised. This feature is extremely useful for taking the load off the controllers early in the exercise. It is also valuable for the aggressor forces, whose tactics and actions are more predictable than those of the officers being trained.

   b) CATTS allows the input of up to ten "Preplanned Missions", each comprised of a variable number of command and control events. The only restriction is that a total of one-hundred or fewer events for all ten missions is allowed. These preplanned missions have the advantage that the time at which they are to occur is under the dynamic control of the controllers, which provides considerably more flexibility. It is also feasible to have a generic preplanned air strike and modify only the target point each time it is used (using the Change Air

Mission feature of the Command and Control System), since this allows the controller control over both time and target, yet requires less interaction than planning the air mission from scratch each time.

4) The controllers are given three capabilities with regard to command and control of air units.

   a) A new mission/unit may be created from scratch.

   b) An existing mission may be cancelled.

   c) An existing mission may be changed.

5) An "air mission" consists of a unit and up to nine associated air route points. Each route point consists of an x coordinate, a y coordinate, an altitude, and a speed. One of the route points may also be designated as a target, which may be a ground unit, an x,y coordinate, a road, or a bridge. The ground trace of the flight between the route points follows the straight line segments connecting those points on the map. The altitude and speed are adjusted dynamically each 15 seconds until they reach the altitude and speed for the next route point. The adjustments made are within input aircraft performance limits for each aircraft type.

6) The air movement model is a time step model. Because of the high speeds of aircraft, the time step used by air movement is nominally 15 seconds instead of 60 seconds and can be changed by input to be even less than 15 seconds.

7) Because of the shorter time step for air units, each air unit actually has at least four locations per math model time step. The location at the beginning of the time step is also required, making at least five locations. In addition, any air route points reached during the math model time step are saved, since they normally represent a change of direction. Thus, a total of up to eight locations for each air unit is calculated and stored every math model time step. It is necessary to store them all in order to

compute air to ground interactions and also for purposes of display.

8) No interactions between air units are modeled.

9) Air units always locate and identify their assigned targets successfully. The assumption is that there is a FAC and he is able to perform successfully.

10) Air units are unable to attack any target unless specifically designated by the controller through the command and control menu. The exceptions to this are canned and pre-planned air missions, which may have designated targets built in.

11) An air unit is an entity composed of one or more aircraft of the same type, flying in a single formation.

12) All computations use flat earth geometry.

13) No air-to-air detections or engagements are modeled.

14) A tactical fighter, flying at top speed of 1250 knots, travels 38.5 km/min. Even at cruise speed of 520 knots, it travels 16 km/min. This means that in a typical 15-second move it may advance 4 km across the gaming area. Since the range of many air defense weapons is on the order of 750 meters to 1200 meters, the air unit may be out of range at either end of its move, yet directly overfly the weapons. Further, it may be undetectable during overflight. The converse is also true, which affects aerial detection models, particularly during surveillance and reconnaissance missions. The result is that air-to-ground and ground-to-air firing and target acquisition models have to be differentiated from the ground-to-ground models, in that they require greater continuity of action. Ground units look for and shoot at targets only at the end of each move, and the fire continues for an entire minute. If air units behaved similarly, even at 15-second intervals, the

results would be unacceptable. The computational requirements of realistic line-of-sight calculations between air and ground units quickly become unacceptable, since what is needed is a verdict as to whether any point on a 16 km path is intervisible with the ground unit. Therefore, line-of-sight between air and ground units is assumed to exist.

15) Air movement assumes no terrain interaction for air units. Specifically, relief and vegetation do not affect LOS between air units and ground units. Also, it is assumed that the flight plan altitudes as input by the controller are sufficient to clear terrain and vegetation. Thus, the air movement model will fly an air unit at the input altitudes regardless of whether this is in fact possible.

16) An aircraft is an entity described by the following characteristics for each aircraft type. The characteristic is input as a part of the aircraft equipment for each specific aircraft type.

Variable Name	Aircraft Characteristic
ILLMAN	Crew size
ROME(IAC,1)	Maximum load at best modeled air density
ROME(IAC,6)	Maximum load at worst modeled air density (correct value chosen here defines worst density at which aircraft can become airborne)
ROME(IAC,2)	Maximum altitude capability
ROME(IAC,3)	Minimum flight speed capability
ROME(IAC,4)	Cruise speed
ROME(IAC,5)	Maximum speed
ROFE(IAC,3)	Fuel consumption rate at minimum speed
ROFE(IAC,4)	Fuel consumption rate at cruise speed
ROFE(IAC,5)	Fuel consumption rate at maximum speed
ROFE(IAC,1)	Fuel consumption change for losing altitude

Variable Name	Aircraft Characteristic
ROFE(IAC,2)	Fuel consumption change for gaining altitude
ROFE(IAC,6)	$\frac{\text{Fuel consumption at maximum load}}{\text{Fuel consumption at minimun load}} - 1$
ROFE(IAC,7)	$\frac{\text{Fuel consumption at worst air density}}{\text{Fuel consumption at best air density}} - 1$
ROME(IAC,8)	Maximum acceleration/deceleration capability
ROFE(IAC,8)	Maximum climb/dive capability
ROME(IAC,7)	Poorest meteorological visibility in which aircraft may conduct a mission
IAMTE(IAC,1)	Takeoff delay time
IAMTE(IAC,2)	Landing delay time
EQCAPAC	Fuel capacity of aircraft

An air equipment type other than an aircraft is defined by the following attributes:

Variable Name	Equipment Characteristic
IPVCE(IEQ,1-8)	List of up to eight allowable aircraft types on which this equipment may be used
IMINRE(IEQ)	Minimum range at which equipment may be used (to prevent aircraft damage)
IMAXRE(IEQ,1)	Maximum range at which equipment may be used
ROME(IEQ,4)	Minimum altitude at which equipment may be used
ROME(IEQ,5)	Maximum altitude at which equipment may be used
ROME(IEQ,2)	Minimum speed at which equipment may be used
ROME(IEQ,3)	Maximum speed at which equipment may be used
ROME(IEQ,1)	Weight of equipment including standard ammunition load (if any)

Variable Name	Equipment Characteristic
ROFE(IEQ,1-3)	Fraction of personnel within target area killed by this equipment, for each of three personnel vulnerability classes
ROFE(IEQ,4-8)	Fraction of equipment within target area damaged by this equipment, for each of fire equipment vulnerability classes IEQCLS (if any)
IAMTE(IEQ,1)	Number of ammunition type used by this equipment
IAMTE(IEQ,2)	Number of rounds of ammo in a standard load (if any)
IAMTE(IEQ,3)	Rate of fire (for weapons)
IAMTE(IEQ,4)	Number of drops in a single pass (for bombs)
IAMTE(IEQ,5)	Distance between drops

17) The air movement module contains logic to abort air missions under certain conditions. The logic to abort the mission is in SUBROUTINE AIRABORT. Missions are aborted:

   a. When a CANCEL AIR MISSION command is received through the command and control subsystem.

   b. When meteorological visibility falls below an input minimum for the applicable aircraft type, i.e., VISM < ROME (IAC,7) where IAC = aircraft equipment number.

18) When an existing air mission is changed, the location of the aircraft at the time of the change is used as the first of the (up to) 9 route points, and the others are as selected by the controller. Other than this, processing is almost identical to the creation of a new air mission.

19) Each time step density altitude is re-computed from current meteorological conditions and compared against the requirements of each air unit, considering current aircraft load. Aircraft whose requirements exceed current density altitude are attrited.

20) When an air unit runs out of fuel (this is unlikely considering extensive error checking during processing of the event notice creating or changing the air mission) it becomes defunct.

21) Air attacks are executed across the width of the target unit regardless of the direction the air unit happens to be moving immediately before and immediately after delivery. The geometrical computation in subroutine FRFD assumes this in determining the area of overlap between the ordnance lethal area and the target unit's area.

5.8.3 Equations

The important tactical and physical mathematical equations used by the Air Module chains are presented below:

5.8.3.1 AIREVENT Chain

All listed equations are in subroutine AIREVENT.

1) Calculate density altitude

Aircraft performance is significantly affected by the density of the air it flies through. Air density is a function of temperature. Density altitude is the altitude in a standard atmosphere corresponding to a particular value of air density as determined by ambient temperature.

DNSALT = (TEMP - 59.)/5.5 * 304.8

where

DNSALT = density altitude

TEMP = ambient temperature in degrees Fahrenheit

2) Convert knots to meters/minute

This is a straight conversion from knots (nautical miles per hour) to meters per minute.

IV = FLOAT(IV) * CKNT2MSC + .5

where

IV	=	velocity in meters per minute
FLOAT(IV)	=	the floating point value of the velocity in knots
CKNT2MSC	=	30.866 = conversion factor

3) Calculate fuel consumed on a flight leg, as influenced by airspeed

Fuel consumption is a non-linear function of airspeed. In order to simplify the air module an approximation was substituted for the actual fuel consumption curves, as affected by airspeed. The air module uses a two piece linear approximation of fuel consumption vs. airspeed with its calculations based on input values of fuel consumption at minimum speed, cruise speed and maximum speed.

The fuel consumed on a leg of the flight is calculated by adding together: (1) the amount consumed at the lowest airspeed for that portion of the curve; and (2) the amount consumed as a result of flying faster than that airspeed, calculated by linear interpolation.

$$DF = DF + \underbrace{\frac{(IV - ROME(IAC,M1))}{(ROME(IAC,M2) - ROME(IAC,M1))} * (ROFE(IAC,M2) - ROFE(IAC,M1)) * TIME}_{(2)}$$

$$+ \underbrace{ROFE(IAC,M1) * TIME}_{(1)}$$

where

DF	=	amount of fuel consumed to this leg
IV	=	velocity
M1,M2	=	variables set to determine which portion of the fuel consumption curve to use; M1 = 3,4; M2 = 4,5; M2 = M1 + 1
ROME(IAC,3)	=	minimum aircraft speed
ROME(IAC,4)	=	cruise speed
ROME(IAC,5)	=	maximum aircraft speed
ROFE(IAC,3)	=	fuel consumption rate at minimum aircraft speed
ROFE(IAC,4)	=	fuel consumption rate at cruise speed
ROFE(IAC,5)	=	fuel consumption rate at maximum aircraft speed
TIME	=	length of time spent on this leg

4) Calculate pressure density

The pressure density calculations done in AIREVENT are the same as those done in the AIRMOV chain. See AIRMOV equation (2) for the explanation.

5) Calculate maximum load aircraft can carry

The load carrying capabilities of the aircraft are modeled as a linear function of air density. The calculation of the maximum load the aircraft can carry is done as a linear interpolation between the maximum load at best air density and maximum load at worst air density.

```
MAXLOAD = ROME(IAC,1) + PDFAC * (ROME(IAC,6) - ROME(IAC,1))
```

where

MAXLOAD	= maximum load the aircraft can carry at a given air density
ROME(IAC,1)	= maximum load the aircraft can carry at the best pressure density
ROME(IAC,6)	= maximum load the aircraft can carry at the worst pressure density
PDFAC	= pressure density factor

6) Calculate weight of fuel necessary to transport the aircraft load

This calculation is based on the assumption that the weight of fuel in pounds necessary to transport the aircraft load is proportional to the weight of fuel necessary to transport the basic aircraft.

```
DF = FLOAT(NPDSFUEL) * DE * (1 + ROFE(IAC,6))/ROME(IAC,1)
```

where

DF	= weight of fuel necessary to transport the aircraft load
NPDSFUEL	= weight of fuel required for basic aircraft to fly mission
DE	= weight of equipment to be transported
ROME(IAC,1)	= maximum weight of load aircraft can carry
ROFE(IAC,6)	= ratio of fuel consumption at maximum and minimum load

5.8.3.2 AIRMOV Chain

All listed equations are in subroutine AIREVENT.

1) Convert Fahrenheit temperature to Kelvin

$$T_K = (T_F + 40)\,\frac{5}{9} + 233$$

$$= (T_F - 32)\,\frac{5}{9} + 273$$

where

T_F = temperature in degrees Fahrenheit (calculated by weather model)

T_K = temperature in degrees Kelvin

2) Calculate air density factor

$$DFAC = (DBEST - DNSTY)/(DBEST - DWORST)$$

where

DBEST = 288/244 = best air density ratio

DWORST = 288/322 = worst air density ratio

DNSTY = $288/T_K$

T_K = temperature in degrees Kelvin

DFAC = air density factor = fraction of distance between DBEST and DWORST at which DNSTY lies

3) Calculate density altitude

$$DNSALT = (T_F - 59)\,\frac{304.8}{5.5}$$

where

T_F = temperature in degrees Fahrenheit (see equation 1)

DNSALT = density altitude

4) Calculate fuel consumption factor for aircraft load and air density effects

$$\text{FFACTOR} = \left(1 + \frac{\text{LOAD}}{\text{XLOAD}} \cdot \text{LFACTR}\right)\left(1 + \text{DFAC} \cdot \text{DFACTR}\right)$$

where

LOAD = current aircraft load in pounds

XLOAD = maximum aircraft load in pounds (input for each aircraft type)

LFACTR = additional fuel consumption at maximum load

$$= \frac{\text{fuel consumption at maximum load}}{\text{fuel consumption at minimum load}} - 1$$

(input for each aircraft type)

DFAC = air density factor (see equation 2)

$$\text{DFACTR} = \frac{\text{fuel consumption at worst air density}}{\text{fuel consumption at best air density}} - 1$$

FFACTOR = fuel consumption factor for aircraft load and air density effects

5) Calculate aircraft speed

$$V = V \pm \text{MIN}\left(|\text{DV}|,\ \text{DT} \cdot \text{ACC}\right)$$

where

V = current aircraft speed in meters/minute

DV = desired speed for this leg minus current speed (desired speed is input as part of flight route for each mission)

DT = duration of current movement period - nominally 1/4 minute (see equation 18)

ACC = maximum aircraft acceleration capability (input for each aircraft type) in meters/minute/minute

$||$ = absolute value function

MIN = minimum value function

6) Calculate altitude change

$$\text{DZ} = \pm \text{MIN}\left(|\Delta Z|,\ \text{DT} \cdot \text{CD}\right)$$

7) Calculate new aircraft altitude

$$Z = Z + DZ$$

where

Z = current aircraft altitude in meters

DZ = altitude change this movement period in meters

ΔZ = desired altitude for this flight leg (input for each mission) minus current altitude

DT = duration of current movement period - nominally 1/4 minute (see equation 18)

CD = maximum climb/dive capability (input for each aircraf type) in meters/minute

8) Calculate ground distance traveled

$$DG = V \cdot DT$$

where

DG = ground distance traveled this movement period

V = current aircraft ground speed (see equation 5)

DT = duration of current movement period (see equation 18)

9) Calculate new X coordinate

$$X_i = X_{i-1} + DG \cdot COS$$

10) Calculate new Y coordinate

$$Y_i = Y_{i-1} + DG \cdot SIN$$

where

X_i = X coordinate of aircraft at end of movement period i

Y_i = Y coordinate of aircraft at end of movement period i

DG = distance traveled over ground during current movement period (see equation 7)

COS = cosine of direction of travel of air unit

SIN = sine of direction of travel of air unit (see equations 11 and 12)

11) Calculate cosine of direction of travel

$$COS = DX/D$$

12) Calculate sine of direction of travel

$$SIN = DY/D$$

where

DX = X coordinate of next air route point minus X coordinate of air route point just reached, in meters

DY = Same as DX, but for Y coordinate

$D = \sqrt{DX \cdot DX + DY \cdot DY}$ = distance between route point just reached and the next route point, in meters

COS = cosine of direction of travel to next route point

SIN = sine of direction of travel to next route point

13) Calculate pounds of fuel consumed, including aircraft speed and altitude change effects

$$DF = DZ \cdot UPDN + (V - S1) \cdot DT \cdot \frac{(F2 - F1)}{(S2 - S1)} + F1 \cdot DT$$

where

DZ = altitude gained or lost this movement period (see equation 6)

UPDN = fuel consumption factor for losing altitude if $DZ < 0$ or fuel consumption factor for gaining altitude if $DZ > 0$ (input for each aircraft type)

V = current aircraft speed (see equation 5)

S1 = aircraft minimum speed if $V <$ cruise speed, or cruise speed, if $V >$ cruise speed

S2 = aircraft cruise speed if $V <$ cruise speed, or maximum speed, if $V >$ cruise speed (minimum, cruise, and maximum speeds input for each aircraft type) all in meters/minute

F1 = fuel consumption in pounds/minute at speed S1 (input for each aircraft type)

F2 = same as F1 for speed S2

DT = duration of current movement period - nominally 1/4 minute (see equation 18)

DF = fuel consumption for this movement period, excluding load and air density effects

14) Update aircraft fuel load

$$FUEL = FUEL - \Delta F$$

15) Update aircraft payload

$$LOAD = LOAD - \Delta F$$

where

FUEL = current aircraft fuel load in pounds

LOAD = current aircraft load in pounds, including fuel, ammo. and equipment

ΔF = pounds of fuel consumed this movement period (see equation 16)

16) Calculate actual pounds of fuel consumed

$$F = MAX(DF,DT \cdot F3) \cdot FFACTOR$$

where

DF = fuel consumption this movement period (see equation 12 in pounds

DT = duration of this movement period - nominally 1/4 minute (see equation 18)

F3 = fuel consumption at minimum speed (input for each aircraft type) in pounds/minute

FFACTOR = fuel consumption factor for load and density effects (see equation 4)

17) Calculate time at which end of movement leg reached

$$T_i = T_{i-1} + DT$$

where

T_j = time at which movement leg j for this time step ends

= time at which movement leg j + 1 begins

DT = duration of this movement leg - nominally 1/4 minute (see equation 18) get point each time it is used (using the Change Air

18) Calculate time duration of current movement leg

$$DT = MIN(TNOM, D1/V)$$

where

TNOM	=	nominal movement duration in minutes - set to 1/4
D1	=	distance to next air route point, at which direction of travel will change
V	=	current aircraft speed (see equation 5)
DT	=	duration of current movement leg in minutes

5.8.3.3 <u>AIRMOV2 Chain</u>

The subroutine containing the referenced equation is indicated in parentheses.

1) Calculate slant range from aircraft to ground unit (AIRCAS)

$$R = \sqrt{(X - x)^2 + (Y - y)^2 + Z^2}$$

where

R	=	slant range
(X,Y,Z)	=	coordinates of aircraft
(x,y)	=	coordinates of ground unit

2) Calculate time aircraft exposed to fire of ground unit on single 1/4 minute movement leg (AIRCAS)

$$TEXP = (1 - RANDOM\ (2 - K))\ (T2 - T1)$$

$$TEXP = 0; \quad K = 1 \text{ or } 2; \quad K = 0$$

where

TEXP	=	time exposed
K	=	number of endpoints of movement leg at which air unit is within range of and has been detected by ground unit. Detection verdicts calculated by GRNDAIR.
T2	=	time at which aircraft reached end of movement leg (see equation 17, Section 5.8.3.2)
T1	=	time at which aircraft began movement leg (see equation 17, Section 5.8.3.2)

3) Calculate number of rounds fired by air defense weapon (AIRCAS)

$$\text{FIRE} = \text{MIN}(\text{TEXP} \cdot \text{ROF} \cdot \text{NUM} \cdot (1 - \text{SIGU}), \text{NROUNDS})$$

where

TEXP =	time aircraft exposed to fire, in minutes (see equation 2)
ROF =	rate of fire of weapon, in rounds/minute - input for each air defense weapon
NUM =	number of pieces of equipment currently operable and manned in ground unit - calculated by REDIST
SIGU =	suppression level of ground unit - calculated by SUPRES
NROUNDS =	number of rounds of appropriate ammunition type remaining in ground unit
MIN =	minimum value function
FIRE =	number of rounds fired

4) Update current ammo level (AIRCAS)

$$\text{NROUNDS} = \text{NROUNDS} - \text{FIRE}$$

where

NROUNDS =	number of rounds of applicable ammo type remaining in unit
FIRE =	number of rounds fired (see equation 3)

5) Update personnel vulnerability classes for units with air defense weapons (AIRCAS1)

$$\text{PERNVC}_{i,j,K} = \text{PERNVC}_{i,j,K} + \text{NUMP} \cdot \text{NUME}$$

where

$\text{PERNVC}_{i,j,K}$ =	number of personnel in unit i with personnel vulnerability class j and suppression mode K
NUMP =	number of personnel required to man air defense weapon (calculated by MANNING)
NUME =	number of pieces of air defense weapon currently manned in unit (calculated by REDIST)

6, 7) Transform range and deflection errors into X, Y aim point errors for air-delivered ordnance (ADW)

AIMERX = RERROR · COS - DERROR · SIN

AIMERY = RERROR · SIN + DERROR · COS

where

RERROR, DERROR =	normally distributed range and deflection errors for this ordnance and delivery conditions (standard deviation of errors calculated in equations 20 to 29)
SIN, COS =	sine and cosine of aircraft flight path angle (see equations 11 and 12, Section 5.8.3.2)
AIMERX, AIMERY =	X and Y aim point errors

8, 9) Calculate nominal impact point for air-delivered ordnance (ADW)

XIM = AIMX + AIMERX

YIM = AIMY + AIMERY

where

AIMX, AIMY =	X, Y coordinates of aim point (input by controller)
AIMERX, AIMERY =	calculated X, Y aim point errors (see equations 6, 7)
XIM, YIM =	nominal X, Y impact coordinates

10, 11) *Calculate individual aim points for air ordnance types with* multiple drops per pass (ADW)

X_i = XIM + (i - 1 - N/2) · DELTA · COS

Y_i = YIM + (i - 1 - N/2) · DELTA · SIN

where

(XIM, YIM) =	calculated nominal impact point (see equations 8, 9)
N =	number of drops per pass (input for each ordnance type)
DELTA =	spacing of drops (input for each equipment type)
SIN, COS =	sine and cosine of aircraft flight path angle (see equations 11, 12, Section 5.8.3.2)
i =	drop number ($1 \leq i \leq N$)
(X_i, Y_i) =	calculated individual ordnance impact coordinates

12-15) Calculate boundaries of lethal area rectangle for air ordnance (ADW)

$$XMIN = XI - PTLNTH/2$$

$$XMAX = XI + PTLNTH/2$$

$$YMIN = YI - PTWDTH/2$$

$$YMAX = YI + PTWDTH/2$$

where

XMIN, XMAX =	minimum and maximum extent of lethal area in X direction
YMIN, YMAX =	minimum and maximum extent of lethal area in Y direction
PTLNTH, PTWDTH =	length and width of lethal area (see equations 43-46)
XI, YI =	coordinates of single air ordnance impact point (see equations 10, 11)

Note: (XMIN, YMIN) and (XMAX, YMAX) define the coordinates of the "lower left" and "upper right" corners of a rectangle.

16) Calculate personnel casualties in a ground unit resulting from air ordnance impact (ADW)

$$C = \sum_{i=1}^{N} OVRLAP \cdot FRAC_i \cdot FIRRATE \cdot P_i$$

where

OVRLAP =	fraction of unit area intersected by ordnance lethal area. (calculated by FRFD)
$FRAC_i$ =	fraction of personnel in vulnerability class i killed by ordnance when within lethal area (input for each ordnance type and vulnerability class in variable ROFE)
FIRRATE =	calculated number of rounds or pieces of ordnance to fall in target area (calculated stochastically in ADW using input fire rate and probability of hit for each ordnance type)
P_i =	number of personnel in vulnerability class i (calculated by ground fire module using mode distribution vectors)
N =	number of personnel vulnerability classes (input)

C = number of personnel casualties resulting from ordnance impact

17) Calculate number of equipment casualties resulting from a single air ordnance impact (ADW)

$$C = OVRLAP \cdot FIRRATE \cdot FRAC \cdot T$$

where

OVRLAP, FIRRATE, C = (same as equation 16)

FRAC = fraction of equipment class 1 killed when within lethal area (input for each air ordnance and ground equipment type in variable ROFE(IBETA,4)

T = total number of pieces of this equipment type curren in unit.

18) Convert speed in meters/minute to knots (ADWDATA)

$$V2 = (V1/100)/0.305$$

$$= V1 \cdot \frac{60}{6000} \cdot \frac{1}{0.305}$$

where

V1 = speed in meters/minute

V2 = speed in knots

0.305 = number of meters in one foot

60 = number of minutes in one hour

6000 = approximate number of feet in one nautical mile

19) Convert meters to feet (ADWDATA)

$$X2 = X1 \cdot \frac{1}{0.305}$$

where

X1 = distance in meters

X2 = distance in feet

20, 21) Calculate standard deviation of range and deflection delivery errors for air ordnance classes 1 to 5 (general purpose bombs), 6 to 8 (high drag bombs) and 17 (napalm) (ADWDATA)

$$RPE = \sqrt{(8FT)^2 + (C1 \cdot SR^2/Z)^2}$$

$$DPE = \sqrt{(8FT)^2 + (C2 \cdot SR)^2}$$

where

FT =	length of time required for ordnance to fall to ground
C1 =	input constant = 0.0212
C2 =	input constant = 0.0184
SR =	slant range to target at ordnance release point (see equation 35)
Z =	aircraft altitude (see equation 7 in Section 5.8.3.2)
RPE, DPE =	standard deviations of range and deflection delivery errors

22, 23) Same as 20, 21 except for equipment classes 11 and 12 (2.75 in. rockets and 20 mm cannon, respectively) (ADWDATA)

RPE = input constant for each class

DPE = input constant for each class

24, 25) Same as 20, 21 except for equipment class 13 (CBU-2)(ADWDATA) Equations are identical to 20, 21 except:

$C1 = 0.0163$; $C2 = 0.0171$

26, 27) Same as 20, 21 except for equipment class 10 (Maverick and other guided projectiles) (ADWDATA)

$RPE = 0.0$

$DPE = 0.0$

28, 29) Same as 20, 21 except for equipment classes 14, 15 (CBU-24 and Rockeye) (ADWDATA)

$RPE = T(IALT, IVEL) \cdot C$

$DPE = RPE$

where

C =	input constant = 0.674
T(IALT, IVEL) =	table look-up function for altitude level IALT and speed level IVEL at delivery (see equations 30 and 31)

30, 31) Calculate altitude level and speed level for use in table look-up functions (ADWDATA)

$$IALT = I, \text{ if } |A(I) - Z| < |A(J) - Z|$$

for every $J \neq I$

$$IVEL = K, \text{ if } |V(K) - S| < |V(L) - S|$$

for every $L \neq K$

where

Z = aircraft altitude at delivery (see equation 7 in Section 5.8.3.2)

S = aircraft speed at delivery (see equation 5 in Section 5.8.3.2)

A = table of "break" altitudes, in feet

= 300, 500, 750, 1000, 1500, 2000, 3000

V = table of "break" velocities, in knots

= 250, 350, 400, 450, 500, 600

Note:

$1 \leq I \leq 7$

$1 \leq K \leq 6$

32) Calculate time required for ordnance to fall to ground for equipment classes 1 to 5 and 17 (GP bombs and napalm)(ADWDATA)

$$FT = \sqrt{2 \cdot Z/G}$$

where

Z = aircraft altitude in feet (see equation 7 in Section 5.8.3.2)

G = gravitational constant

= 32 ft/sec^2

FT = fall time in seconds

33) Same as 32 for equipment class 13 (CBU-2)(ADWDATA)

$$FT = 0.31(Z)^{0.54}$$

where

Z = aircraft altitude in feet (see equation 7 in Section 5.8.3.2)

FT = fall time

34) Same as 32 for equipment classes 6 to 8 (high drag bombs) (ADWDATA)

$$FT = T(IALT, IVEL, ITYPE)$$

where

$IALT, IVEL$ = altitude and velocity levels (see equation 30, 31)

$ITYPE$ = high drag ordnance type (1, 2, or 3)

T = table look-up function

FT = fall time

35) Calculate aircraft slant range at ordnance release point for equipment classes 1 to 5, 6 to 8, 13, and 17 (ADWDATA)

$$SR = \sqrt{Z \cdot Z + R \cdot R}$$

where

Z = aircraft altitude in feet

R = ground range in feet (see equations 36 to 38)

SR = slant range in feet

36) Calculate ground range at ordnance release point for equipment classes 1 to 5 and 17 (GP bombs and napalm) (ADWDATA)

$$R = FT \cdot V \cdot C$$

where

FT = time required for ordnance to fall to ground (see equations 32 and 33)

V = aircraft speed in knots

C = input constant to convert knots to ft/sec = 1.667

R = ground range in feet

37) Same as equation 36 for equipment classes 6 to 8 (high drag bombs) (ADWDATA)

$$R = T(IALT, IVEL, ITYPE)$$

where

IALT, IVEL = altitude and velocity levels at release point (see equations 30 and 31)

ITYPE = ordnance type = 1, 2, or 3

T = table look-up function

R = ground range in feet

38) Same as equation 36 for equipment class 13 (CBU-2)(ADWDATA)

$$R = 615(Z)^{0.101} + 1.3 \cdot V - 700$$

where

Z = aircraft altitude in feet

V = aircraft speed in knots

39) Calculate lethal area of air-delivered ordnance for equipment classes 1 to 8 and 17 (ADWDATA)

$$A = 1.0/(T_1(\mathrm{ITYPE}) - T_2(\mathrm{ITYPE}) \cdot \mathrm{ANGLE})$$

where

ITYPE = ordnance type number

T_1, T_2 = table look-up functions of constant values

ANGLE = ordnance impact angle (see equations 49 and 50)

A = lethal area

40) Same as equation 39 except for equipment classes 9 to 12 (ADWDATA)

$$A = T(\mathrm{ITYPE})$$

where

ITYPE = ordnance type number

T = table look-up function of constant values

A = lethal area

41) Same as equation 39 except for equipment class 13 (ADWDATA)

$$A = C + 4 \cdot Z$$

where

C = constant = 700

Z = aircraft altitude at delivery

A = lethal area

42) Same as equation 39 except for equipment class 15 (ADWDATA)

$$A = T(IANGLE)$$

where

T = table look-up function of constant values.

$IANGLE = [(ANGLE + 14)/15]$

ANGLE = ordnance impact angle in degrees (see equations 49, 50)

$[\;]$ = greatest integer function

43, 44) Calculate length and width of lethal area on ground, for air-delivered ordnance with equipment classes 1 to 13 and 15 (ADWDATA)

$$PTLNTH = \sqrt{A/3}$$

$$PTWDTH = 3 \cdot PTLNTH$$

where

A = lethal area (see equations 39 to 42)

PTLNTH = effective lethal length to use for casualty

PTWDTH = effective lethal width to use for casualty computations

Note: $PTLNTH \cdot PTWDTH = 3\sqrt{\frac{A}{3}} \cdot \sqrt{\frac{A}{3}} = A$

45, 46) Same as equations 43, 44 except for equipment classes 14 and 17 (ADWDATA)

$$PTLNTH = T_1(ITYPE)$$

$$PTWDTH = T_2(ITYPE)$$

where

ITYPE = ordnance type

T_1, T_2 = table look-up functions of constant values

PTLNTH, PTWDTH = effective length and width to use in casualty computations

47, 48) Calculate vertical and horizontal velocity components of ordnance at impact for equipment classes 1 to 5, and 17 (ADWDATA)

$$VH = V \cdot COSD$$

$$VZ = V \cdot SIND + G \cdot FT$$

VH, VZ = horizontal and vertical components of ordnance velocity at impact

V = aircraft speed at delivery (input by controller)

SIND, COSD = sine and cosine of dive angle (assumed to be 30^{o})

G = gravitational constant = 32 ft/sec^{2}

FT = fall time of ordnance (see equations 32 to 34)

49) Calculate impact angle for air-delivered ordnance with equipment classes 1 to 5, and 17 (ADWDATA)

$$ANGLE = ATAN(VZ/VH) \cdot C$$

where

VZ, VH = vertical and horizontal components of ordnance velocity at impact (see equations 47, 48)

ATAN = inverse tangent function

C = constant to convert radians to degrees

ANGLE = impact angle in degrees

50) Same as equation 49 except for equipment classes 6 to 8, and 15 (ADWDATA)

$$ANGLE = T(IALT, IVEL, ITYPE)$$

where

IALT, IVEL = altitude and velocity levels of aircraft at delivery (see equations 30, 31)

ITYPE = ordnance type

T = table look-up function of constant values

ANGLE = impact angle in degrees

51) IEND = FIRRATE * IAMTE(IBETA,4) + .01 (ADW)

Total number of projectiles released (IEND) = number released on one drop (FIRRATE) * number of drops + a factor insuring correct floating to integer conversion.

```
52-55)  IXYX(1) = IXY(II,1) + 0.5 * IW    (ADW)
        IXYX(2) = IXY(II,1) - 0.5 * IW
        IXYY(1) = IXY(II,2) + 0.5 * ID
        IXYY(2) = IXY(II,2) - 0.5 * ID
```

where

IW = IUWID(II) - unit II's width

ID = IUDEP(II) - unit's depth

Half the width (depth) is added and subtracted to the unit's center point X-(Y-) coordinate to yield the X-(Y-) boundaries of the target unit.

```
56)  IDCAS = MIN(ROFE(IBETA,1) * IEND, PERNVC(II,1,1) +
                                       PERNVC(II,2,1))        (ADW)
```

Personnel casualties resulting from a "hard bunker" attack equals the number killed per projectile times the number of projectiles, with an upper bound of the total number of personnel in vulnerability classes 1 and 2 of the target unit II.

5.9 COMMAND AND CONTROL MODULE

The CATTS Command and Control Module controls a diverse variety of events and actions, which allows the controllers to input to the simulation the commands and actions of everyone, from individual squad leaders to Mother Nature. Thus, in addition to the "traditional" command and control functions of maneuver, fire, and tasking organization, the CATTS command and control module controls in the same manner other things, not usually thought of as command and control. These include weather, resupply, control measures, pre-constructed alert messages, unit size, unit deactivation (instant removal from a simulation run), air missions, air defense, and preplanned missions (a collection of any of the above control actions which can be repeatedly made to occur at any time during the simulation).

Maneuver, fire, and tasking organization command and control instructions are input to the simulation using 2 different methods: 1) The events processor, and 2) The command and control decision tables (the old MAFIA V command and control). The other command and control items (weather, resupply, control measures, pre-constructed alert messages, unit size, unit deactivation, air missions, air defense, and preplanned missions) can only be input and activated via the events processor. The events processor interactively accepts command and control during a game. The controllers use the command and control switches on the control panel, the graf pens and the menus on the lower 1/3 of the television monitors to input command and control at the initialization of a game and during a game.

Foreground software drives the menus, the control panel, and the graf pens, and converts the command and control to a 64 word event notice[1] which is passed via a disk file to the background command and control software for processing. This foreground software is described in detail in Section 4.3 of the CATTS Trainer Programming Report, NAVTRAEQUIPCEN 73-C-0156-A005. An overview of the interfaces between the math model and the foreground software is presented in Section 3.3 of the same document. The interested reader is strongly encouraged to refer to this document. A specific discussion of the foreground software is not presented in this User's Manual.

Note 1 - Two formats for the 64 word event notices exist, a foreground and background format. These are shown in Figure 5-138.

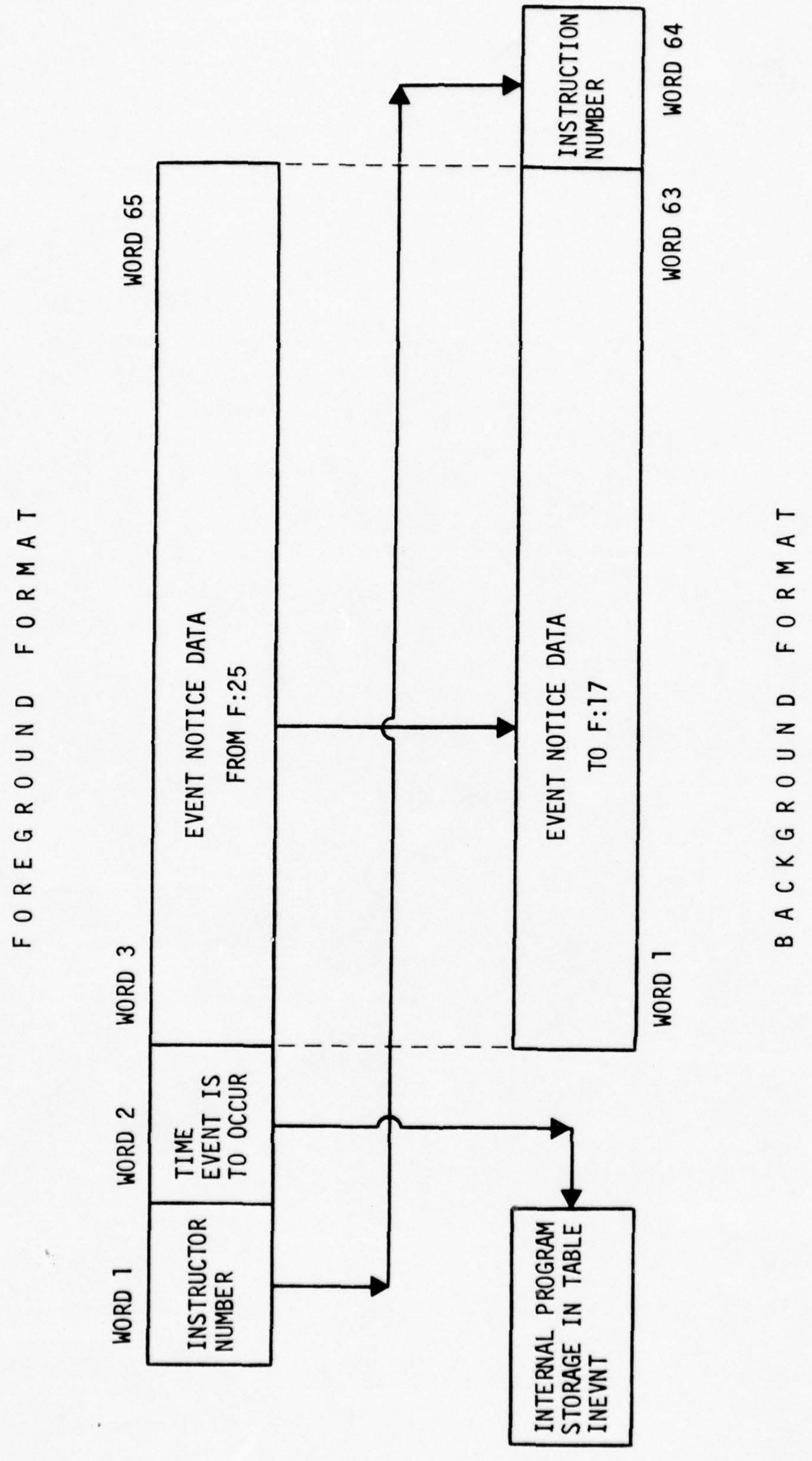

F O R E G R O U N D   F O R M A T
WORD 1
WORD 2
WORD 3
WORD 65
INSTRUCTOR NUMBER
TIME EVENT IS TO OCCUR
EVENT NOTICE DATA FROM F:25
INTERNAL PROGRAM STORAGE IN TABLE INEVNT
EVENT NOTICE DATA TO F:17
INSTRUCTION NUMBER
WORD 1
WORD 63
WORD 64
B A C K G R O U N D   F O R M A T

Figure 5-138. Foreground to Background Event Notice Conversion

The event notices are passed via disk file in order to decouple foreground software and background software, and to preserve data integrity. For example, it would be possible for the foreground software to just go ahead and change model variables directly, rather than go through the seemingly unnecessary complication of passing the data through disk files. However, in order to have the capability to schedule events to occur in the future a disk file is necessary anyway. Additionally, the foreground software is severely core-constrained, making it more attractive to perform the extensive consistency checking needed in the background, rather than the foreground. These reasons would probably be sufficient to dictate the use of disk files. There is an additional issue, however, which makes the use of disk files imperative: data integrity. Two very simple examples will be given to demonstrate the importance of this issue. Both of these examples will assume that the foreground menu software changes math model variables directly, rather than communicating through disk files.

Example 1. Suppose a controller resupplied unit 1/A/2-77 with ten more men. Suppose, moreover, that this occurred in the instant that the math model itself was updating personnel levels for that unit to account for casualties. It is quite possible, in this case, for the math model to store the new, calculated personnel level on top of the level set by the foreground, with the net effect that the ten men never show up in unit 1/A/77.

Example 2. Suppose a controller resupplied all vehicles out of unit 1/A/2-77. Suppose, further, that this occurred during a calculation involving a division by the number of vehicles in the unit. Such divisions normally have a check for zero divisor to protect against division by zero. However, suppose the foreground removed the vehicles <u>after</u> the check for zero, but before the division. The result is that the CATTS exercise would be prematurely and abnormally terminated when the division by zero took place.

These examples may seem farfetched; however, the reader must remember two things. One is that a large number of command and control events are executed during an exercise, and the other is that either of the two results in the examples is completely unacceptable and to be avoided at all costs.

Non-interactive command and control may be input before a game starts via 2 methods: 1) Namelist cards containing 64 word event notices[1] in the run deck or, 2) The prescheduled event disk file. The prescheduled events file can be filled by the user by: 1) Copying punched cards containing 64 word event notices[1] to the file using Xerox RADEDIT or 2) Executing a procedure which saves command and control input during initialization of a game as 64 word event notices[1] on the prescheduled event file (see Appendix B of the Data Base/Operations Manual for the details). All of the command and control input to the simulation via the events processor is activated at the model time input with each command and control event. Thus, events can be scheduled to occur in the future as well as immediately. However, the only thing that activates or triggers events input via the events processor is model time. This is different from what triggers the command and control input via the decision tables (see Table 5-63, Change of State Criteria, in Section 5.9.2.1) for a list of what triggers changes of op states.

The command and control decision tables are input and stored on the data base scenario file (like file NBIG). They are read in during initialization and thus cannot be changed during the game. The information in the condition word of each table is searched by the math model each minute (if events during the minute warrant - not all of the tables are searched each minute), and if a match is found, then the command and control contained in the associated action word of the table is activated. The method of changing the data in these tables is contained in the Data Base/Operations Manual. A more detailed description of the tables and their operation is contained in Section 5.9.3.

Note 1 - See footnote 1 on page 5-885.

Obviously, command and control input via the events processor and the decision tables can conflict with each other. However, a maneuver command given to a unit via the maneuver menu cannot be changed by any of the decision table command and control. The tasking organization control afforded by the decision tables is very weak and unwieldy, but it could counteract tasking organization input via the menus.

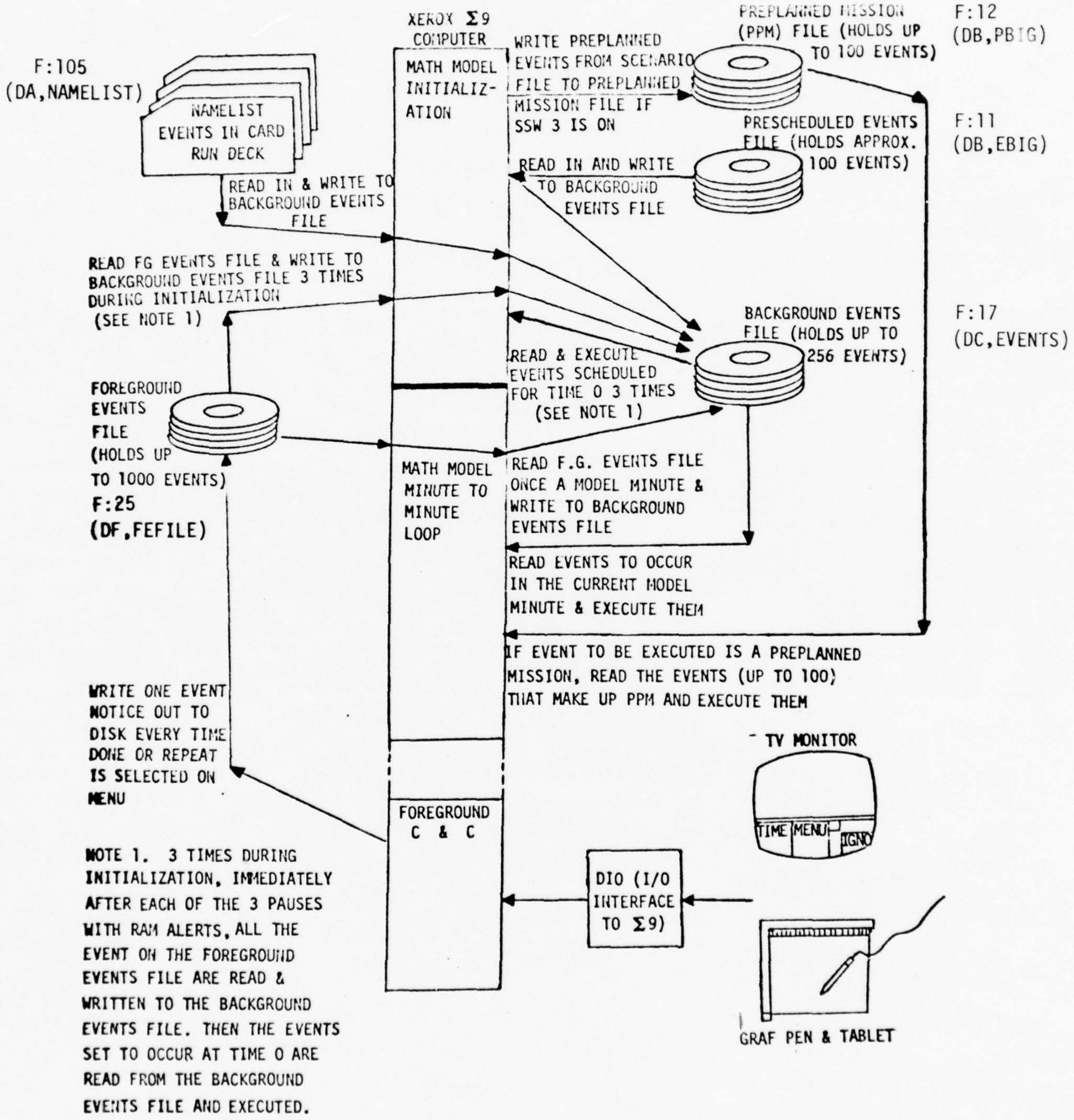
The capability afforded by the decision tables is so powerful and so flexible that the user can tremendously improve the realism of the simulation by implementing changes and additions. (This is particularly true because the tables are delivered to Fort Leavenworth contained only entries of least possible number and complexity). However, the great flexibility provided makes the task of predicting specific outcomes, or conflicts with other data entries, nearly impossible. TRW's experience is that the data tables must be modified with care and "tuned" as experience reveals omissions or conflicts.

In addition, the simulator is designed so that command and control communication with a unit via the menus is temporarily lost when that unit's command post is destroyed. When this destruction happens, all command and control input for the units which are controlled by the devastated command post is ignored for NODEADCP minutes. This decision is made by the ISDEADCO subprogram based on values stored in table IMDEADCP by subroutine STEP. Assumptions made by this subroutine (which are unnecessary and easily removed) prevent the occurrent of communications loss more than once per unit per simulation.

5.9.1 Events Submodule

5.9.1.1 Operation

Figure 5-139 shows all the parts of the events submodule and the general information flow. Figures 5-140 and 5-141 show the subroutine linkages, and Table 5-60 gives a brief description of each subroutine with major inputs and outputs. During initialization, all the events from the prescheduled events file and the run deck are read in by subroutine INPUT and written to the background events file by subroutine PEVENT. Also, if SSW3 is on so that the data base scenario file (NBIG, for example) is being read in, the preplanned mission events are written


XEROX Σ9 COMPUTER
MATH MODEL INITIALIZ-ATION
MATH MODEL MINUTE TO MINUTE LOOP
FOREGROUND C & C
F:105 (DA,NAMELIST)
NAMELIST EVENTS IN CARD RUN DECK
READ IN & WRITE TO BACKGROUND EVENTS FILE
WRITE PREPLANNED EVENTS FROM SCENARIO FILE TO PREPLANNED MISSION FILE IF SSW 3 IS ON
PREPLANNED MISSION (PPM) FILE (HOLDS UP TO 100 EVENTS)
F:12 (DB,PBIG)
PRESCHEDULED EVENTS FILE (HOLDS APPROX. 100 EVENTS)
F:11 (DB,EBIG)
READ IN AND WRITE TO BACKGROUND EVENTS FILE
READ FG EVENTS FILE & WRITE TO BACKGROUND EVENTS FILE 3 TIMES DURING INITIALIZATION (SEE NOTE 1)
BACKGROUND EVENTS FILE (HOLDS UP TO 256 EVENTS)
F:17 (DC,EVENTS)
READ & EXECUTE EVENTS SCHEDULED FOR TIME 0 3 TIMES (SEE NOTE 1)
FOREGROUND EVENTS FILE (HOLDS UP TO 1000 EVENTS)
F:25 (DF,FEFILE)
READ F.G. EVENTS FILE ONCE A MODEL MINUTE & WRITE TO BACKGROUND EVENTS FILE
READ EVENTS TO OCCUR IN THE CURRENT MODEL MINUTE & EXECUTE THEM
IF EVENT TO BE EXECUTED IS A PREPLANNED MISSION, READ THE EVENTS (UP TO 100) THAT MAKE UP PPM AND EXECUTE THEM
WRITE ONE EVENT NOTICE OUT TO DISK EVERY TIME DONE OR REPEAT IS SELECTED ON MENU
TV MONITOR
TIME
MENU
IGNO
DIO (I/O INTERFACE TO Σ9)
GRAF PEN & TABLET

NOTE 1. 3 TIMES DURING INITIALIZATION, IMMEDIATELY AFTER EACH OF THE 3 PAUSES WITH RAM ALERTS, ALL THE EVENT ON THE FOREGROUND EVENTS FILE ARE READ & WRITTEN TO THE BACKGROUND EVENTS FILE. THEN THE EVENTS SET TO OCCUR AT TIME 0 ARE READ FROM THE BACKGROUND EVENTS FILE AND EXECUTED.

Figure 5-139. Events Submodule Operation

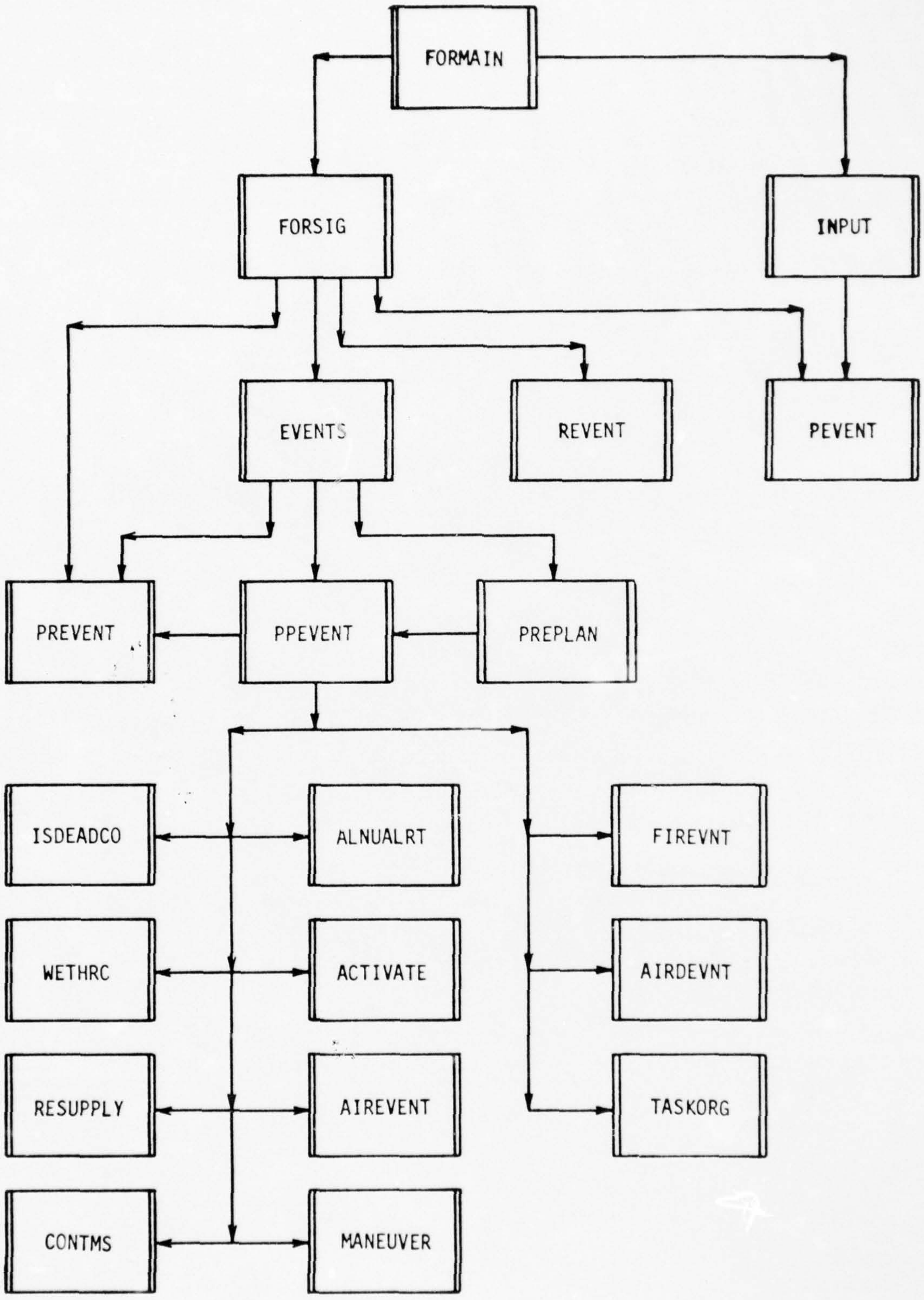


Figure 5-140. Upper Level Subroutine Linkage of Events Submodule

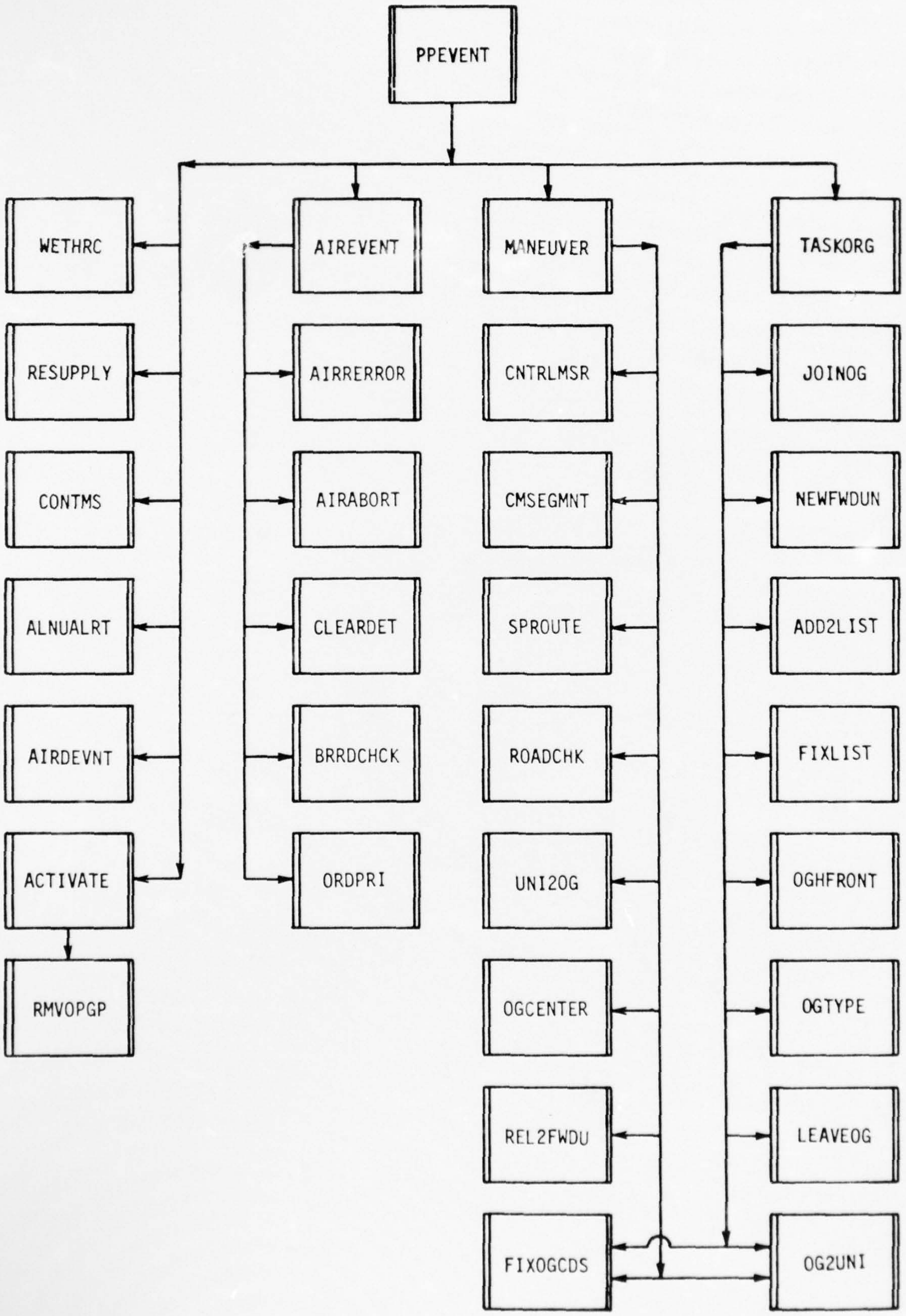


Figure 5-141. Lower Level Subroutine Linkage of Events Submodule

Table 5-60. Events Submodule Subroutine Descriptions

Subroutine	Description	Principal Inputs/Outputs
ACTIVATE	Processes unit activation events (type 7).	Input: INDVDT - 64 word event notice data block. See sample for event type 7. LALL(IJ) - Unit number of the first red (J = 1) and blue (J = 2) units in the list of all units. LISTUN(IU) - A byte-packed table which forms a linked list. For each unit (operational group) IU, the unit immediately following IU is described by: Byte 0 (left-most)--Platoon (and below) list. Byte 1--Company list. Byte 2--Battalion (and above) list. Byte 3 (right-most)--List of all units. Note that $1 \le IU \le 50$. $1 \le IU \le 99$ $101 \le IU \le 120$ Then IU - 100 is an operational group number. $121 \le IU \le 135$ IU is a red unit not modeled but referenced to for convenience.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ACTIVATE (Cont'd)			$136 \leq IU \leq 150$	IU is a blue unit not modeled but referred to for convenience.
				For example, a division control measure must be assigned to a division, even though none are modeled. Therefore "imaginary" units have been introduced for convenience and added realism.
		Output:	ICM(L,M)	- Byte-packed table defining control measures. All measures belonging to deactivated units are deleted.
			ISTATU(IU)	- Status code for unit IU. Set to -1 for all deactivated units.
			LOSIND(IU)	- Table indicating whether line of sight calculations are necessary for unit IU. Set to 0 (no LOS) for deactivated units.
ADD2LIST	Called by TASKURG to add new operational groupings into the list of units and operational groupings.	Input:	I	- Number of dummy "unit" to be added to list = 100 + number of operational group being created.
			KSIZE	- Flag indicating size of group I - corresponds to which of the 3 "unit" lists represented by LISTUN is to be updated to include I.
			ICOLOR	- 1 if I is red, 2 if blue.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ADD2LIST (Cont'd)		Input:	NEXTHI	- Number of the "unit" which is the next higher command to the new operational group.
		Output:	LBATLN(J)	- Numbers of the first red (J = 1) and blue (J = 2) "units" in the battalion size list of LISTUN.
			LCOMPNY(J)	- Same as LBATLN, except for company size.
			LPLTLN(J)	- Same as LBATLN, except for platoon size.
			LISTUN	- (See subroutine ACTIVATE).
			NOALL(J)	- Total number of red (J = 1) and blue (J = 2) "units" in the list of all "units" of LISTUN.
			NOBATLN	- Same as NOALL, except for battalion size.
			NOCOMPNY	- Same as NOALL, except for company size.
			NOPLTN	- Same as NOALL, except for platoon size.
AIRDEVNT	Processes air defense events (type 14).	Input:	INVDT	- (See subroutine ACTIVATE). Also see sample in text.
			IAIRDFLG(IU)	- Air defense flag for unit IU. 1 = Fire at will 2 = Fire if attacked 3 = Do not fire } Air defense weapons

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIRERROR	Creates RAMALERT for errors encountered by subroutine AIREVENT in processing air mission event notices (type 8).	Input: IERROR - Error number encountered. IU - Number of unit to which error applies. IEQ - Number of equipment type to which error applies (if any). UNNAME(J,IU) - First 4 (J = 1) and last 4 (J = 2) characters of the name of "unit" IU. $1 \le IU \le 99$ IU is unit. $101 \le IU \le 120$ IU - 100 is op group. $121 \le IU \le 135$ IU is "imaginary" red unit. $136 \le IU \le 150$ IU is "imaginary" blue unit. Note that IU is defined identically for LISTUN (See subroutine ACTIVATE). INVDT - (See subroutine ACTIVATE plus sample in text.) Output: None

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT	Process air mission events (type 8) to create, delete, or change an air mission. Extensive error checking is performed in addition to creation of all parameters necessary to define an air mission.	Input: IAMTE(IEQ,IMODE) - Ammunition type used by equipment type IEQ operating in mode IMODE, for ground equipments. For aircraft equipment types: IMODE = 1 Takeoff delay time for aircraft, in minutes. = 2 Landing delay, in minutes. For air advance equipment types: IMODE = 1 Number of ammunition type required, if any. = 2 Number of rounds in a standard load of the required ammo type, if any. = 3 Rate of fire, in rounds/minute. = 4 Number of drops per pass (for bombs). = 5 Distance between drops in meters. = 6 Dud percentage. = 7 Kill percentage for bridge type 1.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: IAMTE(IEQ,IMODE) (Cont'd) = 8 Kill percentage for bridge type 2. IEQCOD(IEQ) - Equipment category code for equipment type IEW. -3 = Air sensor. -2 = Air weapon. -1 = Aircraft. 0 = Ground equipment other than weapon. 1 = Direct fire ground weapon. 2 = Indirect fire ground weapon. 3 = Support fire ground weapon. 4 = Air defense ground weapon. INVDT - (See subroutine ACTIVATE, also sample in text). IPVCE(IEQ,IMODE) - Personnel vulnerability class for equipment type IEQ operated in mode IMODE, for ground equipment only. For air equipment IPVCE(IEQ,1) through IPVCE(IEW,8) specify the equipment types of up to eight aircraft for which equipment IEQ is a valid selection.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: IPVCE(IEQ,IMODE) (Cont'd) The first zero encountered means there are no more allowable aircraft for equipment IEQ. NFDU - Number of the first unit index which is available for use as an air unit. By convention $NRU < NFDU < NLDU < NRUP1$. (That is, air units are located between red and blue units in the unit arrays). Used to calculate the correspondence between air unit array index and ground unit array index (air unit 1 is ground unit NFDU). NLDU - Same as NFDU, except last available unit index. Note that $NLDU - NFDU < 10$ is required because of array size limitations. ROFE(IEQ,IMODE) - Rate of fire of equipment type IEQ operated in mode IMODE, for ground equipment <u>only</u>. For air ordance: IMODE = 1, 2, 3 Fraction of personnel in ground unit who are in personnel vulnerability class IMODE and within the target area

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: ROFE(IEQ,IMODE) (Cont'd) who will be killed by equipment IEQ. IMODE = 4, 5, 6, 7, 8 Fraction of equipment with IEQCLS = IMODE -3 which will be damaged by equipment IEQ when within target area. For aircraft: IMODE = 1 Fuel expenditure change for losing altitude in pounds/meter. = 2 Same as 1 for gaining altitude. = 3 Fuel expenditure rate at minimum speed, minimum load, minimum density altitude. = 4 Same as 3, except for cruise speed. = 5 Same as 3, except for maximum speed. = 6 Ratio of fuel expenditure rate at maximum load to that at minimum load.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: ROFE(IEQ,IMODE) (Cont'd) = 7 Ratio of fuel expenditure at highest density altitude to that at lowest density altitude. = 8 Not used. ROME(IEQ,IMODE) - Desired rate of movement for equipment type IEQ operated in mode IMODE, in meters/minute, for ground equipment <u>only</u>. For aircraft: IMODE = 1 Maximum load of aircraft in pounds. = 2 Maximum altitude of aircraft in meters. = 3 Minimum speed of aircraft in meters/minute. = 4 Cruise speed. = 5 Maximum speed. = 6 Maximum load aircraft can carry, in pounds, at highest modeled density altitude. The correct negative value will ensure

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: ROME(IEQ,IMODE) (Cont'd)
		that the aircraft will not be able to fly at density altitudes above its realistic capabilities.
		= 7 Poorest meteorological visibility, in meters, in which aircraft may continue its mission.
		= 8 Not used.
		For other air equipment:
		IMODE = 1 Weight of equipment, in pounds, including standard ammunition load.
		= 2 Minimum aircraft speed at which equipment may be used, in meters/minute.
		= 3 Same as 2, except maximum speed.
		= 4 Minimum altitude, in meters, at which equipment may be used.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Input: ROME(IEQ,IMODE) (Cont'd) = 5 Same as 4, except maximum altitude. = 6, 7, 8 Not used for sensors. For ordnance, road crater radius when used against roads of type IMODE - 5. Output: IACPV(IPT,IAU), IACPX(IPT,IAU), IACPY(IPT,IAU), IACPZ(IPT,IAU) } Desired speed (X, Y), coordinates, and altitude, respectively, for air mission IAU ($1 \le IAU \le 10$) at point IPT ($1 \le IPT \le 9$) on its route. IPT = 1 is always the takeoff point. The first point such *that IACPZ ≤ 0 is the landing* point. Note IAU = IU - NFDU + 1, where IU is the index into ground unit arrays. INOG(IU) - Operational grouping to which unit IU belongs (or 0 if more) when IU is a ground unit. For air units INOG(IU) = 1 specifies blue. IUNNAME - (See subroutine AIRERROR).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
AIREVENT (Cont'd)		Output: ITEQU(IU,JEQ) - Equipment number of the JEQth equipment belonging to unit IU ($1 \le JEQ \le 14$). For air units the aircraft type is always stored as ITEQU(IU,1) by convention. TOTEQU(IU,JEQ) - Number of pieces of equipment type denoted by ITEQU(IU,JEQ) which belong to unit IU. NTGTPT(IAU) - Number of route point for air mission IAU at which target is located (strikes only). NTGTYP(IAU) - Code specifying target type for air mission IAU. $1 \le NTGTYP(IAU) \le 99$ specifies ground unit number to be attacked. = 201 Specifies a road is target. = 202 Specifies a bridge is target. = 203 Specifies an (X, Y) coordinate is target. ITRAV(IU) - Travel code of unit IU - must be equal to 4 for any active air unit. IOPSTU(IU) - Operational state of unit IU, for ground units. For air units specifies a recon mission (= 1) or an air strike (= 2).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
AIREVENT (Cont'd)		Output:	NETU(IU)	- Number of equipment types belonging to unit IU. Maximum value for index JEQ in definition of ITEQU above.
CONTMS	Processes control measure events (type 3) to add or delete a control measure. See sample event in text.	Input:	INVDT	- 64 word event notice. See sample in text.
			MTYPOFCM(I)	- A byte-packed conversion table to convert between foreground control measure types and background types. If KT is the control measure type contained in the event notice, then the KTth byte from the beginning of MTYPOFCM will contain the background control measure type. The background type numbers are organized in a fairly complicated way. There is only space for 40 different types of control measures in some of the associated data tables; however, the background type obtained from MTYPOFCM has been notified to distinguish two types of firing control measures from movement control measures. Movement measures retain the normal 1 to 50 type indices. Firing measures which produce alert messages when fired across have 64 added to their type numbers. Firing control measures which produce alerts when not fired across have 128 added to their type numbers.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
CONTMS (Cont'd)		Input: MTYPOFCM(I) (Cont'd) The table NAMECM contains a twelve character name for each, control measure type (from 1 to 40). The table LABELCM contains a four character generic name such as "zone," "line," "area," etc., which is added to each 12 character name to produce a total of 16 characters. The proper index into LABELCM is obtained from table ICMBREAK. Thus these tables are all highly interdependent, and caution must be used in modifying them. Output: ICM(J,IC) - Packed table which contains fifteen words of information (J = 1, 15) describing each control measure IC ($1 \leq IC \leq 100$) A complete description of this table is lengthy, and can be found in the nomenclature list. NOCMT - Last entry of table ICM which contains valid data.
CRDLIC	Originally created to process event notices to split existing ground units into two units. No menu was ever developed for this capability, hence CRDLIC was never checked out. CRDLIC was also not updated to reflect additional requirements as CATTS changed over time. In fact, CRDLIC has been deleted entirely from more recent versions of CATTS.	

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
DISTCNK	Calculates distance from a given point to a given line segment. This distance is compared to a given distance and a flag set to indicate the outcome. If the calculated distance is less than the given distance, then it is also returned to the calling routine.	Input: (IX1,IY1) (IX2,IY2) - Two sets of (X, Y) coordinates defining a line segment L. (IX3,IY3) - (X, Y) coordinates of a point whose distance from the line setment L is needed. MAXDIST - Maximum distance from L to P which is of interest. Output: IFLAG - 1 if P not within MAXDIST of L, = 2 if it is within MAXDIST. IDIST - Actual distance from P to L, in meters. Calculated <u>only</u> when IFLAG = 2.
EVENTERR	Called by subroutines RESUPPLY and MANEUVER to create RAMALERT error messages. When certain invalid conditions are detected during event notice processing.	Input: IERR - Number of error encountered. NAMEVENT(I,IEVT) - Twelve character $(I = 1, 3)$ name for each event type IEVT $(1 \leq \text{IEVT} \leq 16)$. Output: RAMALERT
EVENTS	Called once per model timestep to process all events scheduled to occur that timestep. The events are read in from disk (F:17, the background events file) and control is passed to subroutine PPEVENT, except for preplanned mission events, which are passed to subroutine PREPLAN for processing.	Input: INEVNT(I) - Half-word packed table which forms a linked list. The left half-word contains the simulation timestep at which the I event is to occur; the right half-word contains a pointer to the next event entry in INEVNT scheduled to occur after event I. The entire notice for event I is stored on disk in the Ith location on the background events file, F:17.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
EVENTS (Cont'd)		Input:	NEXTEV	- A pointer which identifies the entry in INEVNT scheduled to occur next. This pointer is always updated whenever events are added or deleted. A value of zero indicates there are no scheduled events. Note that the event notice data for the next scheduled event is always maintained in cure in the table INVDT.
		Output:	INVDT	- 64 word event notice. See samples in text.
			INEVNT NEXTEV	- (See above). Maintained as events are removed and processed.
FIREVNT	Processes fire control events (type 10). Actual updates are primarily performed by subroutine FIRSORT.	Input:	INVDT	- (See subroutine ACTIVATE). See text for samples.
		Output:	NCC	- Number of fire commands currently in the fire command tables IFIROURD and ITGTLST.
FIRSORT	Called by FIREVNT to perform the bulk of the processing for fire command event notices (type 10).	Input:	INVDT	- (See subroutine ACTIVATE). See samples in text.
		Output:	IFIROVRD(I)	- Packed table containing information regarding the Ith fire command ($1 \le I \le 100$). Byte 0 = Unit number (1 - 100). Byte 1 = Weapon number (1 - 80).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
FIRSORT (Cont'd)		Output: IFIROVRD(I) (Cont'd) Byte 2 = Duration, minutes (1 - 255). Bits 24 - 27 = Subevent type (0 - 8). 0 = Percent of fire, suppression. 1 = Rounds/minute, suppression. 2 = Percent of fire no suppression. 3 = Rounds/minute, no suppression. 4, 5, 6 = Not used. 7 = Cease fire. 8 = Cancel fire mission. Bits 28 - 31 = Number of targets. IMPORD(J) - Four word array of bit flags indicating whether impacting fires just began. If nth bit from left is set, then nth impacting fire in table IXYIM just began.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
FIRSORT (Cont'd)		Output: IMPTEMP(K) - Half-word packed array storing coordinates of Kth fire command against an (X, Y) coordinate. Left half-word = X/4, right half-word = Y/4. Used only for temporary storage. Data later transferred to IXYIM. ISBEVNT - Flag set to event subtype from INVDT(2). See sample in text. ITGTLST(I,L) - Table specifying up to 8 targets for the Ith entry in the IFIROVRD table. Packed as follows: 1 ≤ L ≤ 8. Bits 0 - 9 = Target number. 0 = x, y point. 1 - 99 = Unit number. 101 - 200 = Bridge number + 100. 201 - 700 = Road segment number + 200. Bits 10 - 17 = Duration in minutes. Bits 18 - 31 = Number of rounds to fire, or percentage to fire, at this target.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
FIRSORT (Cont'd)		Output:	ITGTLST(I,L) (Cont'd)	$9 \leq L \leq 10$. Each of the eight bytes here corresponds to one of the (up to) eight targets provided for by $L \leq 8$. If the target is an X, Y point, this byte points to the entry in the impacting fires array IXYIM (or IMPTEMP) which contains the (X, Y) coordinates of the point.
FIXLIST	Called by subroutine TASKORG for each unit maintain the unit list tables whenever an operational group is added, changed, or deleted.	Input:	I	- Number of unit whose operational grouping has changed.
			NEXTHI	- Number of unit which represented the next higher command of unit I before the op group change (from table NXHGCM).
			LBATLN, LCOMPNY, LPLTN	- (See subroutine ADDRLIST).
			LISTUN, LALL	- (See subroutine ACTIVATE).
			NXHGCM(IU)	- Number of "unit" which is next higher command to unit IU. This includes operational groupings and imaginary units. See LISTUN definition for an explanation.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
FIXLIST (Cont'd)		Output: LALL, LBATLN, LCOMPNY, LPLTN, LISTUN - (See "INPUT").
FIXOGCDS	Called to redeploy the units in an operational grouping when the forward most unit changes for some reason.	Input: IOG - Number of operational grouping to be redeployed.
		IFMUN(J) - Number of unit which is "forward most" of operational grouping J. All other units in the op group are deployed relative to this unit, using variable IOGCDS.
		Output: IOGCDS(IU,I) - Desired deployment of unit IU relative to the forward most unit in unit IU's op group (if any).
		I = 1 Distance in meters, in front of "forward most" unit (negative means to rear of forward most).
		I = 2 Distance to the left of the "forward most" unit (negative means to the right), in meters.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
FORSIGI	Called once per simulation timestep performs CAL4, 9 to obtain pointers to completed events on foreground events file (F:25). These are read and transferred to the background events file (F:17), via a call to subroutine PEVENT. Also performs a CAL4, 5 to notify the foreground that a timestep has occurred. In addition, performs all processing for unit relocation events (type 6). Enables subroutine EVENTS to process any other events schedu scheduled for current timestep.	Input: Reads Foreground Events File (F:25).
		Output: IVDT(K) - 64 word table for temporary storage of event notice data being passed to subroutine PEVENT.
		PDIR(IU,I) - Sine (I = 1) and cosine (I = 2) of direction of movement for unit IU.
		IXY(IU,2) - X coordinate (I = 1) and Y coordinate (I = 2) of unit IU location.
		LOSIND(IU) - Byte packed table used to communicate with Environmental module about unit IU.
		Byte 0 = *Flag indicating* whether line of sight calculations are necessary for unit IU this timestep.
		Byte 3 = Number of the terrain data block (1 - 64) in which unit IU is located.
ISDEADCO	Called to determine whether a given unit, which has just received a command and control event, is eligible to receive the event considering a possible temporary loss of communications caused by destruction of command and control headquarters. A RAMALERT is generated if the unit is ineligible.	Input: IIUN - The number of the "unit" for which the event is intended. This can include operational groups. See definition of LISTUN under subroutine ACTIVATE.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
ISDEADCO (Cont'd)		Input:	IMDEADCP(IU)	- Simulation time at which communications will be restored to unit IU. Zero if communications were never lost.
			INVDT	- (See subroutine ACTIVATE).
			NAMEVENT	- (See subroutine EVENTERR).
			UNNAME	- (See subroutine AIRERROR).
		Output:	RAMALERT	- If IIUN suffering communications loss (ISDEADCO = 1).
			ISDEADCO	= 0 if no communications loss.
				= 1 if communications lost.
JOINOG	Called by subroutine TASKORG to add a given unit to a given operational grouping.	Input:	I	- Number of unit to be added to op group IOG.
			IOG	- Number of op group to add unit I to.
			(IFWD,ILAT)	- Desired values for IOGCDS(I,1) and (I,2). (See OUTPUT).
		Output:	IOGCDS	- (See subroutine FIXOGCDS).
LEAVEOG	Deletes a given unit from the operational grouping to which it belongs, if any. Operational group is adjusted as necessary, including deletion if now contains fewer than 2 units.	Input:	IUNIT	- Number of unit to be deleted from its op group.
			INSTR	- Number of controller station from which relevant event received.
			NEXTHI	- Unit which is to be new "next higher command" to unit IUNIT.
		Output:	NXHGCM	- (See subroutine FIXLIST).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
MANEUVER	Processes all ground movement events (type 9).	Input: INVDT - (See subroutine ACTIVATE). Output: IOPSTU - (See subroutine AIREVENT). ITRAV(IU) - Travel code of unit IU. = 1 Avoid engagement. = 2 Engage as necessary. = 3 Seek engagement. = 4 Air unit. MOVECC(IU) - Flag indicating whether unit IU is moving in response to a movement command and control event. 0 = Not under movement command and control. 1 = Under movement control, mounted specified. 2 = Under movement control, dismounted specified. MVTCD, MVDT1, MVDT2, MVDT3 - These tables control the type of movement and destination for each ground unit. They are described in detail in Section 5.5.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
NEWFWDUN	Called to determine the new "forward-most" unit for an operational grouping, when the old "forward most" unit has been removed from that operational grouping by a task organization event notice (type 16).	Input:	IOG	- Number of operational grouping which needs a new "forward most" unit.
			INSTR	- Number of controller console from which command originated.
			TCD DT1 DT2	- (See subroutine MANEUVER).
			TCDOG VDTA1 VDTA2	- Analogue of MVTCD, MVDT1, and MVDT2, except for operational groupings rather than units. See Section 5.5 for a full explanation.
		Output:	FMUN	- (See subroutine FIXOGCDS).
OGCENTER	Called during processing of ground movement events (type 9) to calculate the lateral center of an operational grouping.	Input:	IGRUP	- Number of operational grouping for which calculations are to be performed.
			IOGCDS	- (See subroutine FIXOGCDS).
		Output:	JCNTR	- Average lateral deployment of all units in op'group IGRUP, relative to for[illegible]ard most unit. (Calculated as sum of IOGCDS (IU,2) for IU a member of IGRUP, divided by number of units in IGRUP).
OGHFRONT	Called during processing of a task organization event (type 16) to calculate the half-frontage of a given operational grouping.	Input:	IOG	- Number of operational grouping for which calculations are to be performed.
			IOGCDS	- (See subroutine FIXOGCDS).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
OGHFRONT (Cont'd)		Output:	HLFRN(J)	- Half the FEBA frontage occupied by operational group J, in meters.
OGTYPE	Called during processing of a task organization event (type 16) to determine the "type" of an operational group based on the unit types of units belonging to the group.	Input:	IOG	- Number of the operational group for which a type calculation is required.
			ITYPEU(IU)	- One of twenty unit types to which unit IU belongs. Unit type is a critical parameter in most of the command and control decision logic.
		Output:	IOGTYP(J)	- Characteristic unit type of operational group J.
PARE	Called to unpack the unit list array LISTUN.	Input:	IL	- Number of "unit" for which unpacking is to be done.
			LSIZE	- Flag indicating which of the four lists is to be unpacked from the corresponding byte ($0 \leq \text{LSIZE} \leq 3$).
			LISTUN	- (See subroutine ACTIVATE).
		Output:	N	- Unpacked value from LISTUN, hence number of next "unit" in the list indicated by LSIZE.
PEVENT	Called to enter events by writing them on the background events file (F:17). Event tables and pointers are also updated to reflect new event.	Input:	IVDT	- (See subroutine FORSIGI).
			ITMEVE	- Simulation time at which event represented by IVDT is scheduled to occur.
		Output:	INEVNT } NEXTEV }	- (See subroutine EVENTS).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
PEVENT (Cont'd)		Output:	ITNE	- Simulation time at which next scheduled event is to occur.
POINTGT	Called to find units, bridges, and road segments which lie close enough to (x, y) points selected as targets in a fire control event notice (type 10) for possible damage to be calculated.	Input:	INVDT	- (See subroutine ACTIVATE).
			K	- Pointer to target location in array INVDT.
		Output:	NT	- Flag indicating whether target found (> 0) or not found (= 0).
			LSAVE	- Number of unit, bridge, or road segment found as a target.
PPEVENT	Called by subroutine EVENTS (or by subroutine PREPLAN for preplanned missions) to check for temporary loss of communications due to destruction of a command and control headquarters (where applicable), and to distribute event notices to the correct routines for processing.	Input:	INVDT	- (See subroutine ACTIVATE).
		Output:	None	
PREPLAN	Called whenever a preplanned mission event (type 15) is executed. The event notices which comprise the preplanned mission are read into core one at a time from the preplanned mission file (F:12) and implemented via calls to subroutine PPEVENT.	Input:	INVDT	- (See subroutine ACTIVATE). For preplanned missions, contains number (1 to 10) of preplanned mission to be executed.
			NPPM	- Total number of preplanned missions available for this scenario ($1 \le NPPM \le 10$).
			IPPREC(I)	- Pointer to disk record which contains the first event in preplanned mission number I.
			NPPREC(I)	- Total number of events comprising preplanned mission number I.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
PREPLAN (Cont'd)		Output:	INVDT	- (See INPUT). Data is read by PREPLAN on top of input data, for use by other event processors.
PREVENT	Called to provide, optionally, either debug print and/or a permanent disk record of event notices received during the initialization phase of an exercise, based on the values of input flags.	Input:	IDUMY	- 64 word local storage of event notice data.
			IFLAG	- Flag indicating whether PREVENT is being called to process an event already on the background events file and about to be implemented (IFLAG = 1), or whether an event has just been received from the foreground and is about to be added to the background events file (IFLAG = 2).
			ISCHEDT	- Time at which event is scheduled to occur.
			IOINTRVL(IS)	- Flag controlling whether event notice debug print is required.
		Output:	None.	
REL2FWDU	Called as part of the processing for a ground movement event (type 9) to calculate and set the movement codes and movement data values to ensure that all units in an operational group move properly relative to each other.	Input:	JOG	- Number of operational group to be moved.
			JUDEP(IU)	- Depth of unit IU in meters (depth measured in direction faced by unit).
			DPMVT(J,L)	- Sine (L = 1) and cosine (L = 2) of direction faced by operational group J.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
REL2FWDU (Cont'd)		Output: MVTCD, MVDT1, MVDT2, MVDT3 - These tables control the movement of ground units. They are described in detail in Section 5.5.
RESUPPLY	Implements resupply events (type 2), performing all necessary updates to unit levels.	Input: INVDT - (See subroutine ACTIVATE). Output: CLAVGAS(IU), CLDIES(IU), CLGAS(IU) - Current levels, in gallons, of aviation gas, diesel fuel, and gasoline, respectively, in unit IU. MENNOW(J,IU) - Half-word packed table of personnel levels for unit IU. J = 1 Left half-word = Number of commanding officers. Right half-word = Number of officers. J = 2 Left half-word = Number of enlisted leaders. Right half-word = Number of enlisted men. PERS(IU) - Total number of personnel currently in unit IU. TOTEQU - (See subroutine AIREVENT).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
RESUPPLY (Cont'd)		Output:	USEEQU(IU,K)	- Total number of pieces of Kth equipment type belonging to unit IU which are currently manned.
			NETAMU(IU,M)	- Total amount (in tenths of a round) of the Mth ammunition type belonging to unit IU which currently remain in that unit. (Similar to TOTEQU, except for ammunition rather than equipment). Actual ammunition type is given by NTAMU(IU,M).
			NTAMU(IU,M)	- Number of the Mth ammunition type belonging to unit IU. The quantity is given by NETAMU(IU,M).
REVENT	Subroutine REVENT Has never been used, because there is no menu to drive it. However, it was originally intended to remove and delte events already scheduled--either specific events, specific types of events, or events scheduled for certain times.	Input:	IND	- + 1 if events scheduled to occur on or after simulation time ITM are to be deleted, - 1 is events scheduled to occur on or before ITM are to be deleted.
			ITM	- Simulation time (0 to ?) to use in deleting events (See IND).
			IT1	- Event type to be deleted.
			IT2	- Event subtype to be deleted.
		Output:	INEVNT, NEXTEV	- (See subroutine EVENTS).
			ITNE	- (See subroutine PEVENT).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
RMVOPGP	Removes a given unit from any operational grouping to which it belongs. If this leaves less than two units in the op group, that op group is disbanded and any control measures belonging to the group are deleted.	Input:	IUNIT	- Number of unit to be removed from its operational group (if any).
		Output:	INOG(I)	- Operational group to which unit I belongs. Zero means not a member of any group.
ROADCHK	Determines whether a given unit is traveling on a road. This is determined from the unit's movement code MVTCD (only units moving on a route, movement code 6, may be "on road") and its current distance from the nearest road segment.	Input:	IUNIT	- Number of the unit for which the "on road" determination is to be made.
			MVTCD(IU)	- Movement code of unit IU. See Section 5.5 for a detailed description.
			MXRDIST	- Maximum distance (in meters) from a road that a unit may be and still be considered "on road." User input Necessary because of slight inaccuracies in controller input of unit routes would make it virtually impossible to achieve on road movement otherwise.
			IRDNPTS(K)	- The number of points, or sets of (X, Y) coordinates, used to define the Kth road ($1 \le K \le 50$). If IRDNPTS(K) = N, then road K is composed of N points defining N - 1 contiguous straight line segments.
			IRDSTRT(K)	- The first point in the road point storage arrays (IROADX and IROADY) which belongs to road K ($1 \le K \le 50$). If

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
ROADCHK (Cont'd)		Input: IRDSTRT(K) (Cont'd) IRDSTRT(K) = M, then (IROADX(M), IROADY(M)) is the first point on road K.
		(IROADX(L), IROADY(L)) - (X, Y) coordinates for the L th road point stored ($1 \le L \le 500$). The road to which this point belongs is defined by IRDSTRT and IRDNPTS.
		NROADS - Total number of roads modeled ($1 \le NROADS \le 50$).
		Output: IRDSEGMT - Number of road segment unit IU is traveling on (if any). ($1 \le IRDSEGMT \le IRDNPTS(IRD) - 1$).
		IRD - Number of road IUNIT is traveling on, if any.
SPROUTE	Searches for available space in which to store the data necessary to define a "special route" for ground movement control (event type 9).	Input: NCDDP(K) - Number of points belonging to special route K ($1 \le K \le 50$). A zero indicates route storage not in use.
		INVDT - (See subroutine ACTIVATE). See text for examples.
		Output: IROUTE - Number of route assigned (if any). Then NCDDP(IROUTE) is the number of points, etc.
		IFOUND - Flag indicating whether storage was successful. = 0 No available space; error. = 1 Space found & route stored.

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		

SPROUTE (Cont'd)		Output:	IXPTH(IPT,K)
			IYPTH(IPT,K)
				- (X, Y) coordinates of point number IPT ($1 \leq IPT \leq 7$) on special route K ($1 \leq K \leq 50$). Number of points actually in use is NCDDP(K).

TASKORG	Processes task organization events (type 16). Consistency checking and all necessary updates are performed via a series of calls to other subroutines.	Input:	DPMVT	- (See subroutine RELZFWDU).
			IDPCOD(J)	- Deployment code of operational grouping J. = 0 Not deployed, cannot move. = 1 Not deployed, can move. = 2 Technically deployed. = 3 Actually deployed.
			IOGCDS	- (See subroutine FIXOGCDS).
			MTCDOG MVDTA1	- (See subroutine NEWFWDUN).
			NXHGCM	- (See subroutine FIXLIST).
UNI2OG	Called during processing of a ground movement control event (type 9) to transfer the movement data for a given operational grouping.	Input:	KOG	- Number of operational grouping to set movement data for.
			IFMUN	- (See subroutine FIXOGCDS).
			MVTCD MVDT1 MVDT2 MVDT3 ITRAV	- (See subroutine MANEUVER).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs		
UNI2OG (Cont'd)		Output:	NTCDOG, MVDTA1, MVDTA2, MVDTA3	- (See subroutine NEWFWDUN).
			ITRAVG(J)	- Travel code for operational grouping J. This is the analogue of the travel code for units (See ITRAV, above).
UN2FEB	Called to set up the proper new destination data values for a unit which belongs to an engaged operational grouping.	Input:	IDUM	- Number of unit for which values are to be set.
			DPMVT	- (See subroutine RELZFWDU).
			IFEBAB(K,1)	- (X, Y) coordinates of center of blue FEBA for engagement K ($1 \leq K \leq 12$). See Section 5.6 for a full explanation.
			IFEBAR	- Same as IFEBAB, except for red forces.
			DIREAX(K,1), DIREAX(K,2)	- Define sine and cosine of the engagement axis for engagement K (blue to red direction) See Section 5.6 for a fuller explanation ($1 \leq K \leq 12$).
			MVTCD, MVDT1, MVDT2, MVDT3	- (See subroutine MANEUVER).
		Output:	IX3, IY3	- (X, Y) coordinates of desired new destination.
WETHRC	Called to process weather change events (type 1).	Input:	INVDT	- (See subroutine ACTIVATE).

Table 5-60. Events Submodule Subroutine Descriptions (Continued)

Subroutine	Description	Principal Inputs/Outputs
WETHRC (Cont'd)		Output: IWCLS - Current weather class selected ($1 \le IWCLS \le 11$). Also sends weather alert message via call to ALERTGEN.

to the preplanned mission file (PBIG, for example) by subroutine PPLANINP. Immediately after each of the pauses during initialization, where the halt light comes on and the RAM alerts come up, the foreground events file is read by subroutine FORSIG and all the events on it are added to the background events file, again by subroutine PEVENT. Then, the events scheduled for minute zero are read from the background events file and executed. Thus, no events are executed until after the first halt during initialization. During the running of the game, the foreground events file is read once a model minute, and the format converted, and then the events are immediately written to the background events file. After the foreground events file is empty, the events scheduled for the current minute are read from the background events file and executed.

The logic flow for the principal routines of the Events Submodule is presented as Figures 5-142, 5-143, and 5-144 (subroutines FORSIG, EVENTS, and PPEVENT, respectively). Some routines not flowcharted are straightforward. PEVENT does the bookkeeping necessary to maintain the event list table, INEVENT, when new events are added to the background events file, and also writes the new event on the file. REVENT performs a similar function when events are deleted - however, since there is no menu in CATTS for deleting scheduled events, subroutine REVENT is never executed during the simulation, and could be omitted entirely with no ill effect. PREVENT prints event notices when a print flag is turned on (IOINTRVL(IS) > 9). It also writes the events processed during initialization to F:50 to enable the user to duplicate initial conditions precisely during later exercises, if desired (when IOINTRVL(IS) > 5).

The Events Submodule is discussed at length in the <u>CATTS Trainer Programming Report</u>, Section 5.9, pages 5-503 through 5-600. In particular, that document contains detailed descriptions of the operation of each event processor (such as subroutine AIREVENT) which are not duplicated in this <u>User's Manual</u>.

The command and control menus may be used to place events on the foreground events file any time during initialization or the running of the game (see <u>Operators Manual</u> for specific operation of command and control console buttons and menus). If at any time (during initialization

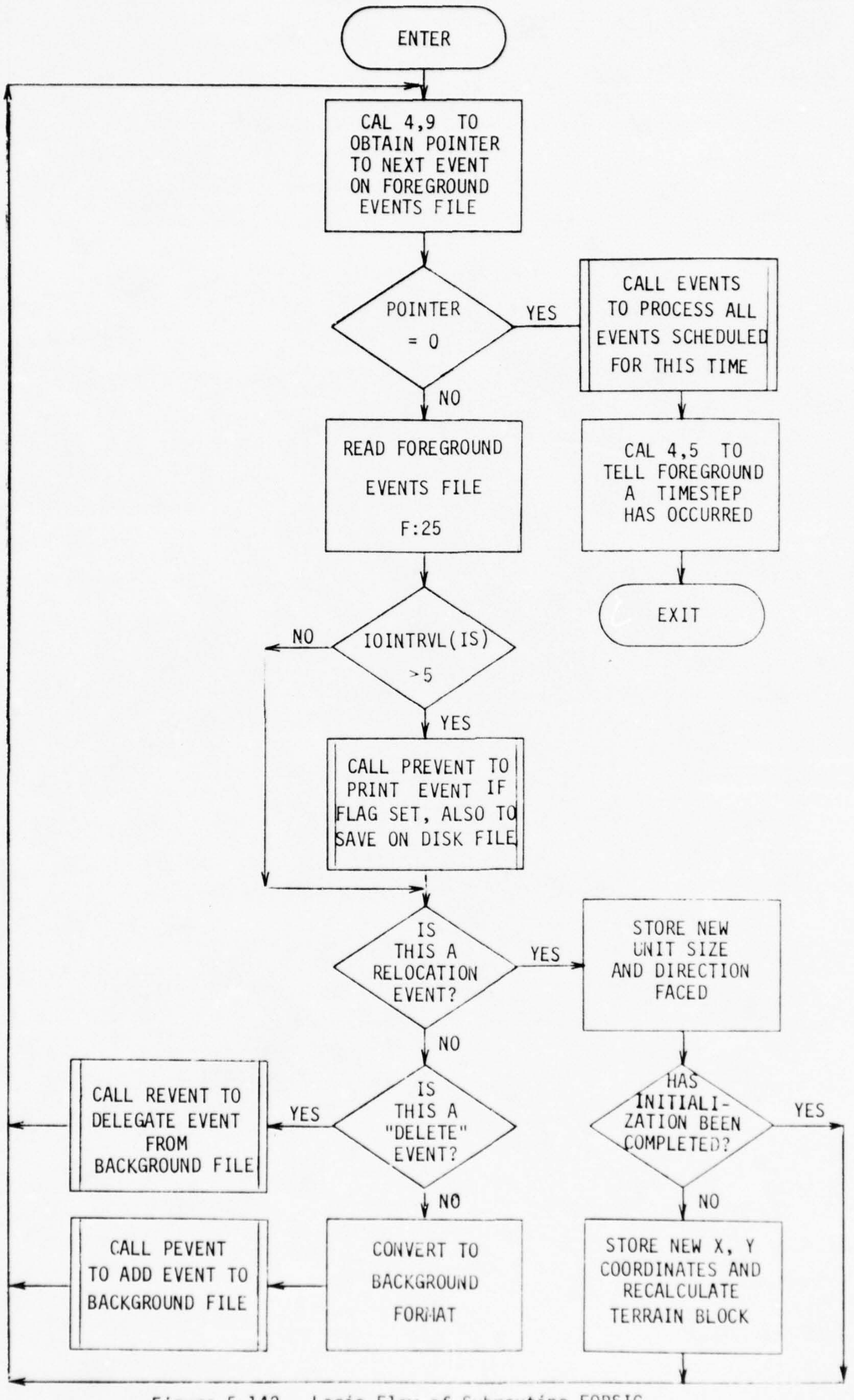

ENTER
CAL 4,9 TO OBTAIN POINTER TO NEXT EVENT ON FOREGROUND EVENTS FILE
POINTER = 0
YES
CALL EVENTS TO PROCESS ALL EVENTS SCHEDULED FOR THIS TIME
CAL 4,5 TO TELL FOREGROUND A TIMESTEP HAS OCCURRED
EXIT
NO
READ FOREGROUND EVENTS FILE F:25
IOINTRVL(IS) >5
NO
YES
CALL PREVENT TO PRINT EVENT IF FLAG SET, ALSO TO SAVE ON DISK FILE
IS THIS A RELOCATION EVENT?
YES
STORE NEW UNIT SIZE AND DIRECTION FACED
NO
HAS INITIALI-ZATION BEEN COMPLETED?
YES
NO
IS THIS A "DELETE" EVENT?
YES
CALL REVENT TO DELEGATE EVENT FROM BACKGROUND FILE
NO
CONVERT TO BACKGROUND FORMAT
CALL PEVENT TO ADD EVENT TO BACKGROUND FILE
STORE NEW X, Y COORDINATES AND RECALCULATE TERRAIN BLOCK

Figure 5-142. Logic Flow of Subroutine FORSIG

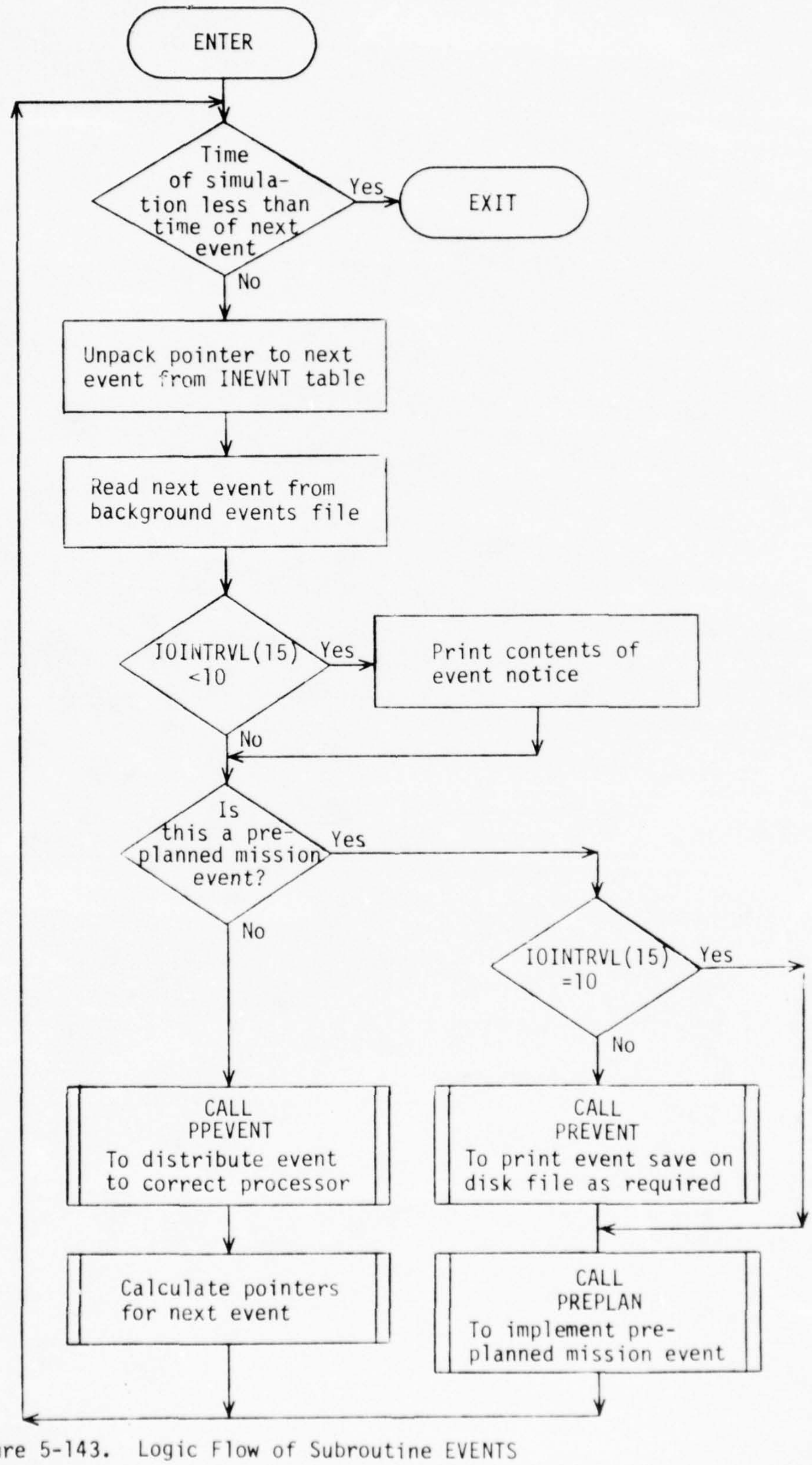

ENTER
Time of simula- tion less than time of next event
Yes
EXIT
No
Unpack pointer to next event from INEVNT table
Read next event from background events file
IOINTRVL(15) <10
Yes
Print contents of event notice
No
Is this a pre- planned mission event?
Yes
No
IOINTRVL(15) =10
Yes
No
CALL PPEVENT To distribute event to correct processor
CALL PREVENT To print event save on disk file as required
Calculate pointers for next event
CALL PREPLAN To implement pre- planned mission event

Figure 5-143. Logic Flow of Subroutine EVENTS

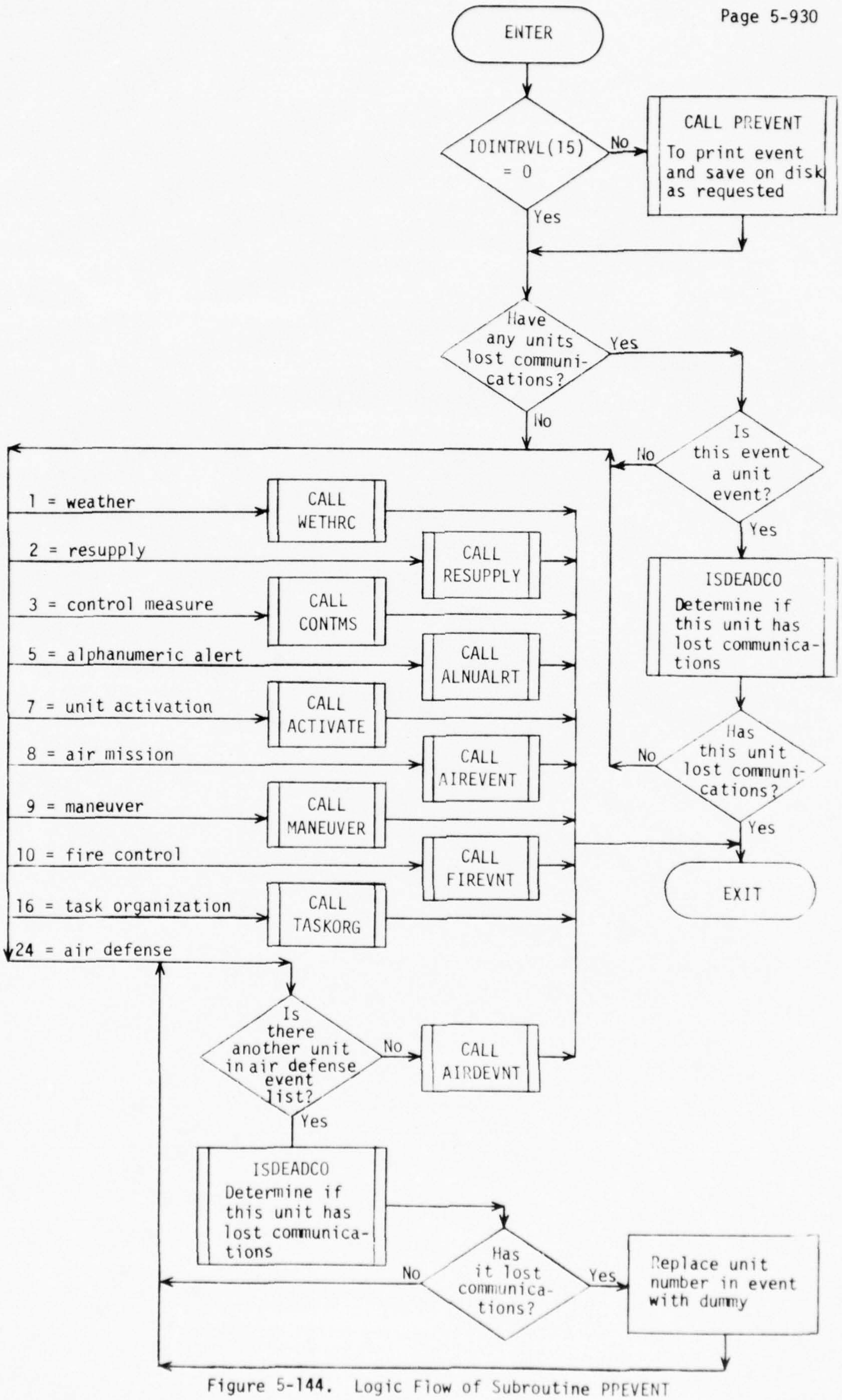


Figure 5-144. Logic Flow of Subroutine PPEVENT

or during a game), a preplanned mission event is executed, the actual events to be executed are read from the preplanned mission file rather than the background events file, although the preplanned mission event itself is kept on the background events file.

Two tables keep track of the events on the background events file; the Event Indicator Table and the Type-Subtype Table. The Event Indicator Table and the Type-Subtype Table are each 250 words long (thus, a maximum of 250 events may be in the Events File at any one time, even though the file is large enough for 256). Once an event occurs, the space occupied by the data for this event becomes available for use by another event. The contents of these two tables are shown in Figure 5-145. The Event Indicator Table contains in the first half of each word, the time (in minutes) at which the event will occur. The second half of each word contains the event number (1-250) of the succeeding event. If this event is the last scheduled event to occur, the event number of the succeeding event is zero. This table is added to every time an event is written to the background events file, and when an event is read from the background events file, the entry corresponding to the event is blanked out for use by newly input events. The table is used by the events submodule to find on disk the proper events to execute during the current minute. Thus, the model knows which events are to occur in a given minute, and the order of the events on the background events disk file. It knows this by using the core stored Event Indicator Table. The Type-Subtype Table contains in the first half of each word the event type, and in the second half of each word the event subtype. This table was intended to allow the deletion of events from the background events file before the events occurred. This capability was never added to the menus as was, at one time, planned. This was due to problems of displaying the current contents of the background events file in a meaningful manner, and the fact that controllers can override old events by putting in a countermanding event. However, the capability still exists in the module, and the user may at some future date wish to make use of it. If so, the following information will be helpful. The routine to delete events (REVENT) from the background events file should be called whenever a scheduled event on the file has not yet occurred. The routine is enterred with the type (IT1) and subtype (IT2) of the event(s) to be deleted, a time (ITM), and an indicator (IND) which is

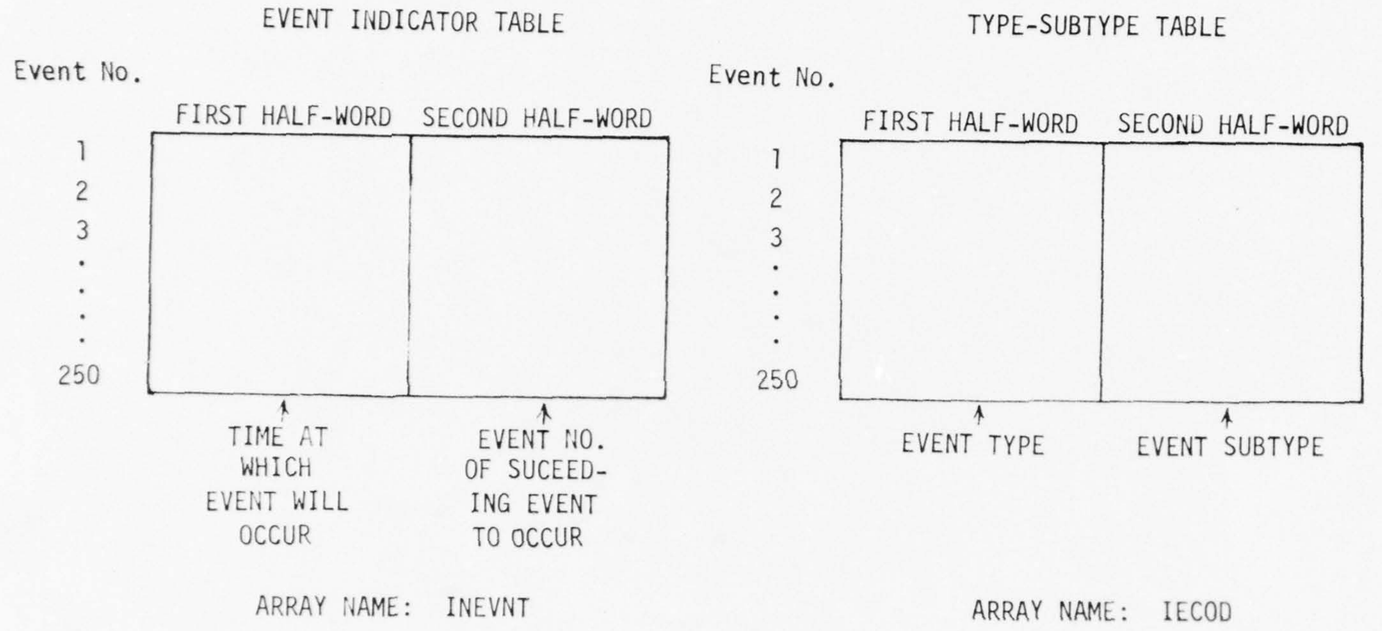


Figure 5-145. Contents of Event Indicator Table and Type-Subtype Table

equal to -1, 0, or +1. The meaning of the indicator is as follows: if IND equals -1, delete all events of the given type and subtype which are scheduled to occur prior to or at time T; if IND equals 0, delete all events of the given type or subtype which are scheduled to occur at time T; and if IND equals +1, delete all events of the given type and subtype which are scheduled to occur after time T. Events are deleted by merely modifying the appropriate entry in the Event Indicator Table.

Table 5-61 shows the various event types, a brief description, and the subroutine which is called by the event submodule to process each of the event types and make them occur.

The formats of the events varies slightly depending on the residence (which of the files or cards) contains the event. However, the exact same information is contained in all the events, by type. The specific contents of each event type, by type, will be covered in the succeeding pages. In general, the events contain a time in minutes at which they are to occur, the instructor # of the instructor console that entered the event (1, 2, or 3), and 63 other information words whose content varies with the event type. The format detailed on the following pages is the form that the events have on the background events file and the preplanned mission file. Notice that no time is needed in this form since the Event Indicator Table keeps track of the time on the background events file and variables IPPREC (starting record number) and NPPREC (# of records in the selected preplanned mission) allow the proper events to be read in whenever a preplanned mission is to be executed. The format on the foreground events file is the same except the instructor # and the occurrence time are stored in prefixed words 1 & 2 respectively, and then foreground word 3 contains the same data shown as word 1, word 4 contains the same data shown as word 2, etc. The format of the events on the prescheduled events file and in the card run deck are identical and are in namelist form. The time of occurrence is in variable ITMEVE, and the 64 data words are variable IVDT, and they correspond exactly word for word with 64 words shown. The examples shown with each event type will be in this format, since this is the format the user must use to input events via punched cards. Debug printout of both the foreground and background form can be obtained using namelistable array IOINTRVL(15). A value of zero means no debug printout, 1-5 means print background format when

Table 5-61. Event Types

Event Notice Type	Description	Event Notice Processor
1	Weather	WETHRC
2	Resupply	RESUPPLY
3	Control Measures	CONTMS
4	SPLIT UNIT (NO MENU)	CRDLIC
5	Generate Alert Message	ALNUALRT
6	Change Unit Location	UNLOCATE
7	Unit Deactivation	ACTIVATE
8	Air Mission	AIREVENT
9	Ground Maneuver	MANEUVER
10	Ground Fire	FIREVNT
11	(Not Used)	
12	(Not Used)	
13	(Not Used)	
14	Air Defense	AIRDEVNT
15	Preplanned Mission	PREPLAN
16	Task Organization	TASKORG

event is activated, >5 means print background format when event is activated and print in foreground format[2] when the event is received from foreground , and ≥9 means print background format when event is activated, print in foreground format when event is received from foreground, and print some additional event debug printout.

The following show the events structure for each type, an example event of the type in namelist form, and a description of what the example event will do.

Note 2 - The foreground uses the area where the event notices are constructed as a scratch area to aid in construction of event notices. Thus, spurious numbers may appear in unused (by background) portions of event notices. Since they are ignored by the background, these numbers don't interfere with the operation of the events processor.

EVENT TYPE 1 - Weather

SUBEVENT TYPE 1

Word	Contents
1	Event Type (=1)
2	Subevent type (=1)
3	New Global Weather Class
4-63	Not Used
64	Instructor #

SAMPLE WEATHER EVENT

```
ITMEVE = 4,
IVDT(1) = 1,
IVDT(2) = 1,
IVDT(3) = 2,
IVDT(64) = 1,
*
```

WEATHER EVENT DESCRIPTION

At 4 minutes into the game, the weather class will be changed to 2(clear).

EVENT TYPE 2 - Resupply

Word	Contents
1	Event Type (=2)
2	Subevent Type 1 = standard units, 2 = % of basic load
3	Unit Number of Recipient
4	Unit Number of Donor
5	Number of CO's to be Transferred
6	Number of Officer to be Transferred
7	Number of Enlisted Men in Leadership Positions to be Transferred
8	Number of Enlisted Men to be Transferred
9	Number of Equipment Resupplies
10	First Equipment Resupply Word Resupply Word:
.	| Equipment Type | Amount Transferred |
.	
.	
23	Fourteenth Equipment Resupply Word
24	Number of Ammo Resupplies
25	First Ammo Resupply Word Ammo Resupply Word
.	| Ammo Type | Amount Transferred |
.	
.	
38	Fourteenth Ammo Resupply Word
39	Amount of Gasoline Transferred
40	Amount of Diesel Transferred
41	Amount of Air Fuel Transferred
64	Instructor #

SAMPLE RESUPPLY EVENT

```
ITMEVE = 0,
IVDT = 2, 1, 20, 27, 0, 1, 4, 7, 1, 655364, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 655520, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 2,
*
```

RESUPPLY EVENT DESCRIPTION

At time 0, the following will be transferred from unit #27 to unit #20:

1	officer
4	enlisted leaders (NCO's)
7	enlisted men
4	equipment type 10 (T-62 tanks)
160	ammo type 10 (115 MM shells)

EVENT TYPE 3 - Control Measures

Word	Contents
1	Event Type (=3)
2	Subevent Type =1 => add line or area
	=2 => delete control measure
	=3 => error
	=4 => add point
3	Type of Control Measure (If subevent Type 2 - delete, this is the number of the control measure (1-100))
4	Unit number of Unit to which Measure Belongs
5	Red or Blue =1 red control measures
	=2 blue control measures
6	X Coordinate of first point
7	Y Coordinate of first point
8	X Coordinate of second point
9	Y Coordinate of second point
10	X Coordinate of third point
11	Y Coordinate of third point
12	X Coordinate of fourth point
13	Y Coordinate of fourth point
14	X Coordinate of fifth point
15	Y Coordinate of fifth point
16	X Coordinate of sixth point
17	Y Coordinate of sixth point
18	X Coordinate of seventh point
19	Y Coordinate of seventh point
20	X Coordinate of eighth point
21	Y Coordinate of eighth point
64	Instructor #

SAMPLE CONTROL MEASURE EVENT

```
 ITMEVE = 0,
 IVDT = 3, 1, 17, 1, 1, 57770, 25572, 59309, 25594,
       -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 2,
*
```

CONTROL MEASURE EVENT DESCRIPTION

At time 0, a line of type 17 (route of march) will be created. It will be a red control measure, associated with unit #1 (for size purposes). It will go from X,Y 57770, 25572 to 59309, 25594.

SECOND SAMPLE CONTROL MEASURE EVENT

```
 ITMEVE = 10,
 IVDT(1) = 3,
 IVDT(2) = 2,
 IVDT(3) = 4,
 IVDT(64) = 3,
*
```

SECOND CONTROL MEASURE EVENT DESCRIPTION

At 10 minutes into the game, control measure #4 will be deleted. This slot (4th) then is available for new control measures.

EVENT TYPE 4 - Create DLIC's (Detachments Left in Contact)

Note: No Menu Currently Exists for this Type of Event, and the CATTS model has been modified to ignore this Type of Event.

Word	Contents
1	Event Type (=4)
2	Event Subtype (=1)
3	Unit Number of First Unit to be Broken up to Create First DLIC
4	Unit Number of First Dummy Unit to be come First DLIC
5	Fraction of Personnel to be put in First DLIC (Floating Point Number)
6	Fraction of Crew Served Equipment to be put in First DLIC (Floating Point Number)
.	
.	
.	
39	Unit Number of Tenth Unit to be Broken up to Create Tenth DLIC
40	Unit Number of Tenth Dummy Unit to become Tenth DLIC
41	Fraction of Personnel to be put in Tenth DLIC (Floating Point Number)
42	Fraction of Crew Served Equipment to be put in Tenth DLIC (Floating Point Number)
43 - 63	Not Used
64	Instructor #

No sample for event type 4 is given as this capability has never been used. The existing code in subroutine CRDLIC has never been checked out and has been removed from the most recent versions of CATTS.

EVENT TYPE 5 - Generate Alert Message

Word	Contents	
1	Event Type (=5)	
2	Routing:	1 = INS 1
		2 = INS 2
		3 = INS 3
		4 = INS 1 & 3
		5 = INS 2 & 3
		6 = INS 1 & 2
		7 = INS 1, 2, & 3
3	First Word of Message	
	.	
	.	
	. (2 lines of 72 characters each)[1]	
30	Last Word of Message	
64	Instructor #	

Note 1: See problem report #335 (only one line is output correctly).

SAMPLE GENERATE ALERT MESSAGE EVENT

```
ITMEVE = 0,
IVDT = 5, 1, -472259008, -488054045, -975904192,
 -942024988, -680836927, 1088866518, -741087168,
 -689515067, -1002380572, -1025324604,
 -910042648, 1087817280, -472259008,
 -908737717, 1799411673, -974993963,
 -471703488, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
*
ITMEVE = 0,
IVDT = 5, 1, -1052712221, -690362304, -691650349,
 -689584789, 1086440677, -975838236, -641373376,
 -1042955200, -473572903, -909978683, 1088864729,
 455144000, 1549556800, -960118079, 1086838483,
 -1002415012, 1547714624, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
```

GENERATE ALERT MESSAGE EVENT DESCRIPTION

At time 0, the following 2 alert messages will appear on the Super Bee terminal at console #1.

```
TRW SYSTEMS GROUP, A WHOLLY OWNED SUSIDIARY OF TRW INC., PRESENTS -
A STORY OF LOVE, ADVENTURE, AND TERRIBLE WAR *** FEBA GOLD ***
```

The "*** FEBA GOLD ***" portion of the second message will blink in reverse background (See Super Bee Manual).

EVENT TYPE 6 - Relocate Units

Word	Contents
1	Event Type (=6)
2	Unit Number
3	SINE of Direction that Unit will Face
4	Unit's New X Coordinate
5	Unit's New Y Coordinate
6	COSINE of Direction That Unit Will Face
7	New Unit Depth
8	New Unit Width
64	Instructor #

SAMPLE RELOCATE UNIT EVENT

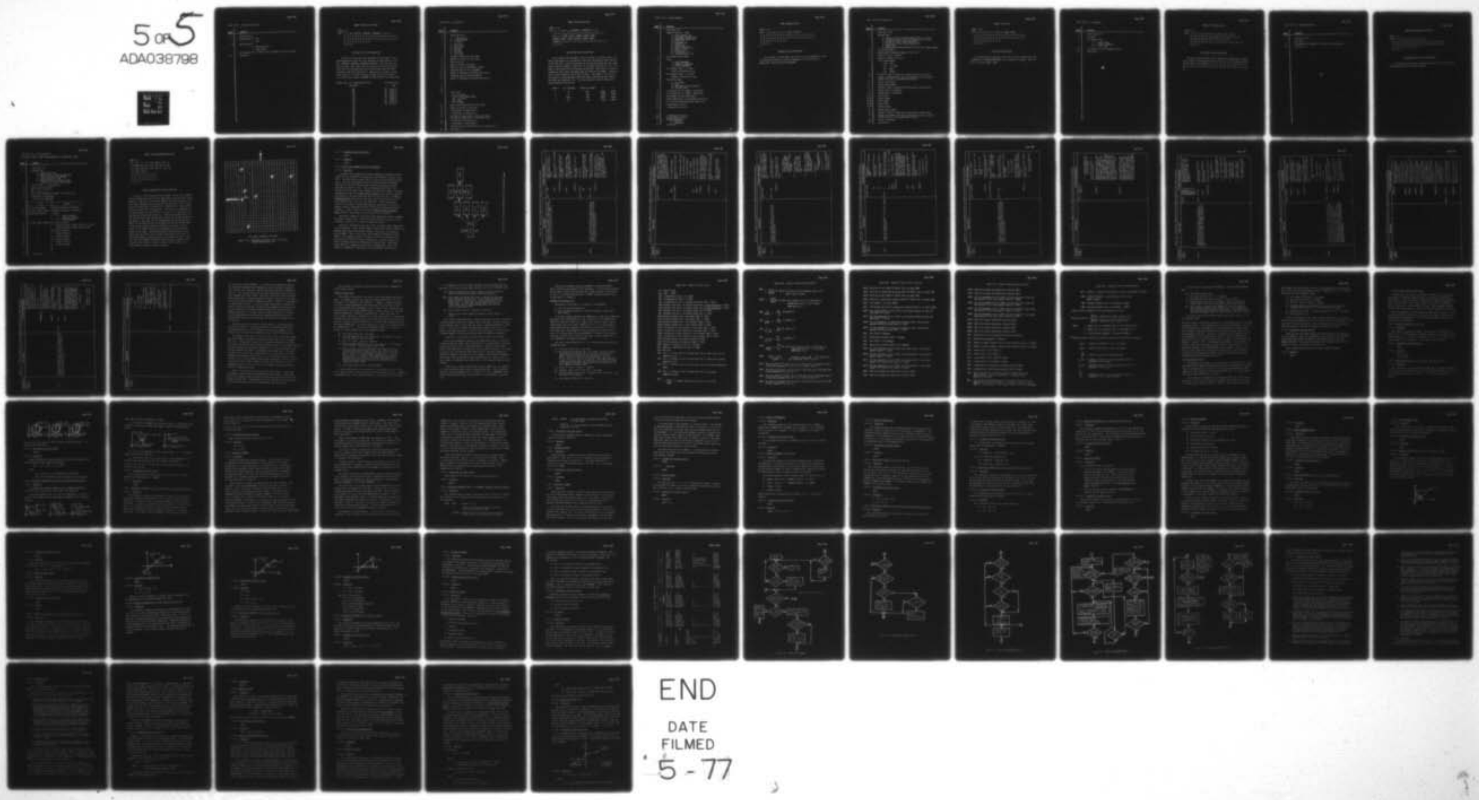
```
ITMEVE = 0,
IVDT = 6, 87, 1077346607, 65650, 18600, -1090127198,
       50, 300, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1,
*
```

RELOCATE UNIT EVENT DESCRIPTION

At time 0, unit #87 will be relocated to X,Y 65650, 18600 facing 282°.4 compass degrees. The units new depth will be 50 meters and new width will be 300 meters. At any time other than during initialization, the only part of this event notice acted upon by the events processor would be the new width and depth for the unit. No unit is allowed instant relocation during the game.

AD-A038 798 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

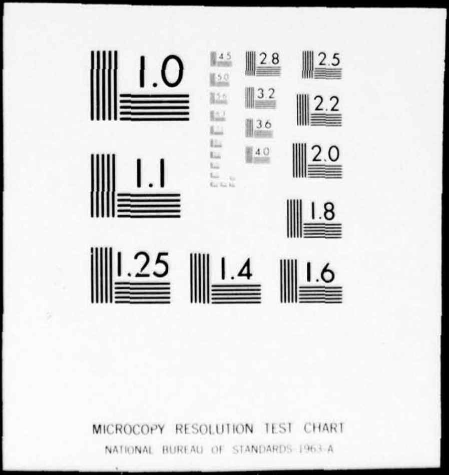
5 OF 5
ADA038798



END
DATE
FILMED
5-77

5 OF 5

ADA038798



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

EVENT TYPE 7 - Select Active Units

Word	Contents
1	Event Type (=7)
2	Subevent Type 0 = red
	1 = blue
3 - 28	Byte Indicators
	0 = deactivate unit
	1 = leave alone
	one byte per unit. Defunct units and dummy units are skipped.
64	Instructor #

SAMPLE ACTIVATE UNIT EVENT

```
ITMEVE = 0,
IVDT = 7, 1, 257, 16777217, 16842752, 16843008, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 3,
*
```

ACTIVATE UNIT EVENT DESCRIPTION

At time 0, 9 blue units were selected to be activated. The 4 byte packed words (3rd, 4th, 5th, 6th words of event notice) in Hex are: 00000101, 01000001, 01010000, 01010100. Thus, the 3rd, 4th, 5th, 8th, 9th, 10th, 13th, 14th, and 15th units in the LISTUN array after the first non-fake blue unit in the LISTUN array have been selected as the active blue units. Shown below, in order, are the units, their #'s, and the first part of the unit list as input to the LISTUN array in deck to CONTROL LISTING OF UNITS in the data base:

PARTIAL UNIT LIST (STARTING WITH FIRST BLUE UNIT)	ACTIVATED UNITS #	NAME
144	49	1/A/2-77
142	50	2/A/2-77
141	51	3/A/2-77
139	56	1/B/2-77
39	63	2/B/2-77
47	57	3/B/2-77
49	74	1/C/2-77
50	82	2/C/2-77
51	79	3/C/2-77
53		
54		
56		
63		
57		
59		
80		
74		
82		
79		
85		
43		
⋮		

EVENT TYPE 8 - Air Mission

Word	Contents
1	Event Type (=8)
2	1: if new mission 2: if modification 3: if cancel
3	1: blue recon 2: blue strike 3: not used 4: not used 5: red recon 6: red strike
4	Equipment number (aircraft type)
5	Number of aircraft in the flight
6	Name of mission
7	Name of mission (continued)
8	Number of types of equipments loaded
9	Equipment type of first equipment
10	Number of equipments associated with word 9
11	Equipment type of second equipment
12	Number of equipments associated with word 11
.	.
.	.
.	.
27	.
28	.
29	Target Type 00 - Not target 1-100 - Unit target number 201 - X-Y Point 202 - Road 203 - Bridge
30	Route point associated with the target
31	Number of points in the route
32	X coordinate of route point 1
33	Y coordinate of route point 1
34	Altitude of route point 1 (first half word) Velocity of route point 1 (second half word)
35	X coordinate of route point 2
36	Y coordinate of route point 2
	Continue for each route point up to a maximum of 9.
64	Instructor #

SAMPLE AIR MISSION EVENT

```
ITMEVE = 37,
IVDT = 8, 1, 6, 60, 2, -641350560, -1043736077, 1, 23,
    5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    201, 3, 4, 30000, 30000, 1638850, 59000, 35000,
    1638850, 60500, 25000, 65536300, 30000, 30000,
    1638850, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 2,
*
```

AIR MISSION EVENT DESCRIPTION

At 37 minutes into the game, a new air mission, a red air strike, will begin (the planes will not take off until after the take-off delay time set in the data base - usually 5 minutes, but the mission begins when the count-down to take-off begins). There will be 2 aircraft type 60 (SU - 7s). The name of the mission will be displayed as RED-AIR3. There will be 1 type or ordnance on the mission, equipment type #23 (500 LB SLICK BOMB). There will be 5 of these loaded on each of the 2 planes. The target is an X,Y point. The target is associated with point #3 of the 4 total points that make up the route the planes will fly. The route consists of the following X,Ys and altitudes and velocities.

POINT #	ALT IN METERS	VELOCITY IN KNOTS	X	Y
1	25	450	30000	30000
2	25	450	59000	35000
3	1000	300	60500	25000
4	25	450	30000	30000

EVENT TYPE 9 - Ground Maneuver

Word	Contents
1	Event Type (=9)
2	Subevent Type (OP - Type) =1 =>HASTY DEFENSE (22) =2 =>DELIBERATE DEFENSE (21) =3 =>DELIBERATE AMBUSH (85) =4 =>DELAY (31) =5 =>WITHDRAW (32) =6 =>RECON (17) =7 =>RECON FORCE (12) =8 =>MOVE TO CONTACT (11) =9 =>ATTACK (13) =10 =>DISPLACING (14) =11 =>EXPLOIT/PURSUIT (16)
3	Unit Number Under Control
4	Intent 1 =>SEEK ENGAGEMENT 2 =>ENGAGE AS NECESSARY 3 =>AVOID ENGAGEMENT
5	For Subevent Types 1, 2, and 3 SINE of direction to face
6	For Subevent Types 1, 2, and 3 COSINE of direction to face
7	Destination Type =1 =>X-Y COORD. =2 =>UNIT =3 =>OLD ROUTE (Control Measure) =4 =>SPECIAL ROUTE
8	X Coordinate for X-Y COORD. Destination
9	Y Coordinate for X-Y COORD. Destination
10	Unit Number for UNIT Destination
11	Control Measure No. for OLD ROUTE Destination
12	No. of Points Defining SPECIAL ROUTE (2-9)
13	X Coordinate of Point 1
14	Y Coordinate of Point 1
. . .	
31	X Coordinate of Point 9
32	Y Coordinate of Point 9
31	Mounted/Dismounted =1 =>Mounted =2 =>Dismounted
64	Instructor #

SAMPLE MANEUVER EVENT

```
ITMEVE = 15,
IVDT = 9, 10, 24, 3, 0, 0, 1, 55600, 21200, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 2,
*
```

MANEUVER EVENT DESCRIPTION

At 15 minutes into the game, unit #24 will go into operational state 14 (displacing) and move, avoiding engagement, to X,Y destination 55600,21200. The unit will move mounted.

EVENT TYPE 10 - Ground Fire

Word	Contents
1	Event Type (=10)
2	Subevent Type 0 = Percent of fire, consider suppression as a factor 1 = Rounds per minute, consider suppression as a factor 2 = Percent of fire, ignore suppression 3 = Rounds per minute, ignore suppression **7 = Cease fire** **8 = Delete entire** override entry for fire unit, weapon number
3	Firing unit number
4	Weapon number (equipment number) to be fired
5	Duration of fire in minutes
6	Number of targets to fire at
7	First target number 1-100 UNIT # 201 X,Y POINT 202 ROAD 203 BRIDGE
8	Percent (1-100) or rounds per minute against first target
9	Target X coordinate (not used by background for unit targets - current unit X,Y is used when event occurs)
10	Target Y coordinate
11	Second target number
12	Percent (1-100) or rounds per minute against second target
13	Second target X coordinate
14	Second target Y coordinate
15-18	Third target
19-22	Fourth target
23-26	Fifth target
27-30	Sixth target
31-34	Seventh target
35	Eighth target number
36	Percent (1-100) or rounds per minute against eighth target
37	Target X coordinate (not used by background for units targets - current unit X,Y is used when event occurs)
38	Target Y coordinate
64	Instructor #

SAMPLE FIRE EVENT

```
ITMEVE = 48,
IVDT = 10, 1, 25, 17, 2, 1, 201, 6, 68000, 19000,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 2,
*
```

FIRE EVENT DESCRIPTION

At 48 minutes into the game, unit #2 will fire 6 rounds/minute from equipment #17 (152MM GUN/HOWITZER) for 2 minutes at 1 target. This target is X,Y point 68000, 19000.

EVENT TYPE 14 - Air Defense

Word	Contents
1	Event Type (=14)
2	Not used
3	Air Defense Flag =1 = FIRE AT WILL =2 = FIRE IF ATTACKED =3 = DON'T FIRE
4 - 63	Unit numbers to which command applies
64	Instructor #

SAMPLE AIR DEFENSE EVENT

```
ITMEVE = 0,
IVDT = 14, 0, 1, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
       81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
       95, 96, 97, 98, 99, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1,
*
```

AIR DEFENSE EVENT DESCRIPTION

At time 0, and from then on until another air defense event changes their orders, the following units will fire at will at any enemy aircraft within range: unit numbers 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, and 99.

EVENT TYPE 15 - Preplanned Mission

Word	Contents
1	Event Type (=15)
2	Not Used
3	Preplanned Mission Number of mission to be implemented
64	Instructor #

SAMPLE PREPLANNED MISSION EVENT

```
ITMEVE = 12,
IVDT = 15, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 1,
*
```

PREPLANNED MISSION EVENT DESCRIPTION

At 12 minutes into the game, preplanned mission #2 will be read from the preplanned mission file and executed.

EVENT TYPE 16 - Task Organization

(The term "TEAM" is used interchangeably for operational group)

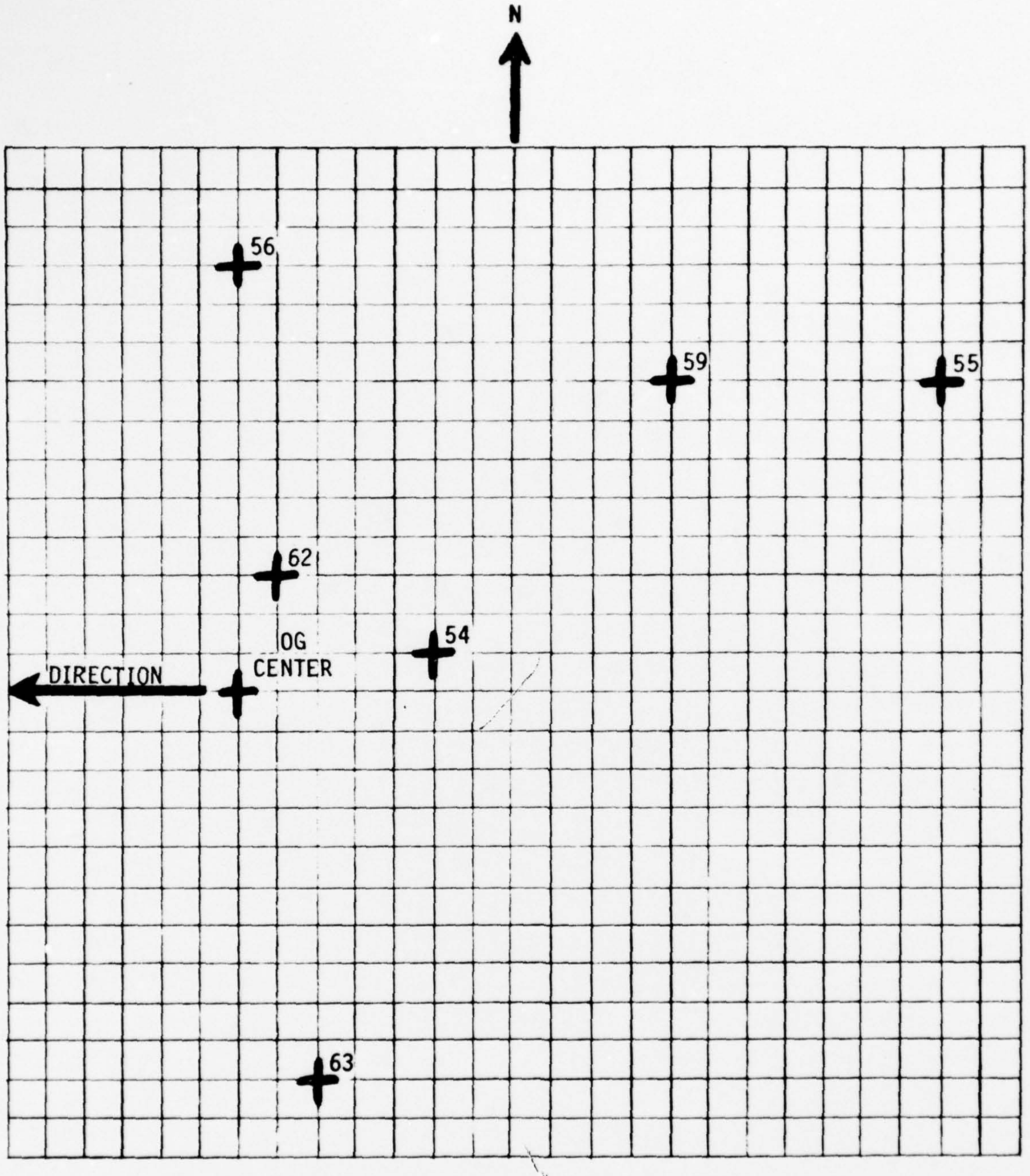
Word	Contents	
1	Event Type (=16)	
2	Subevent Type =1 = CREATE A NEW TEAM =2 = ADD A UNIT TO A TEAM (NO MENU PAGE EXISTS TO SELECT THIS OPTION) =3 = DELETE UNITS IN A TEAM =4 = REDEPLOY UNITS IN A TEAM (NO MENU PAGE EXISTS TO SELECT THIS OPTION)	
3	Size of team: (For CREATE only)	
4	Operational Group Number	
5	No. of units in list	
6	Next higher command unit number (For CREATE only)	
7	X location for deployment	
8	Y location for deployment	
	DELETE	CREATE
9	First unit deleted	SINE of direction of movement
10	His next higher command	COSINE of direction of movement
11	Second unit deleted	Intent: =1 = Avoid Engagement =2 = Engage as necessary =3 = Seek Engagement
12	His next higher command	1st unit affected
13	"	forward distance (-means rearward of center)
14	"	lateral distance (-means to the right)
15	"	2nd unit affected
16	"	forward distance
17	"	lateral distance
18	"	3rd unit affected
19	"	forward distance
20	"	lateral distance
. . .		
64	Instructor #	

SAMPLE TASKING ORGANIZATION EVENT

```
ITMEVE = 0,
IVDT = 16, 1, 1, 6, 6, 136, 72540, 21721, 0,
  -1091567616, 2, 54, -250, -50, 56, 0, -550,
  55, -900, -400, 63, -100, 500, 62, -50, -150
  59, -550, -400, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1,
*
```

TASKING ORGANIZATION EVENT DESCRIPTION

At time 0, a new op group will be created. It will be a company sized op group, and it will use the name input in the data base for op group #6. There are 6 units to be combined into this new group. The next higher command of the op group will be "fake unit" #136. The op group will deploy with its center at X,Y 72540, 21721. Figure 5-145. on the next page will illustrate this. The direction of movement, or facing direction, will be 270° (sin = 0, cos = -1). The units in the op group get travel code (intent) 2. Unit 54 will be 250 meters to the rear and 50 to the right of center, 56 will be 0 meters behind and 550 meters to the right, 55 will be 900 meters to the rear and 400 meters to the right, 63 will be 100 meters to the rear and 500 meters to the left of center, 62 will be 50 meters to the rear and 150 meters to the right and 59 will be 550 meters to the rear and 400 meters to the right. Tasking organization events executed concurrently with the first 2 halts during initialization will result in instant deployment of the op group to the selected locations. After this (the "LAST CHANCE TO RELOCATE UNITS" message is output at the 2nd halt), the units will move at a normal rate of speed in the normal manner to deploy to these locations.



Each square represents 100 meters

Figure 5-145. Deployment Locations of OG 6 From Sample Tasking Organization Event

5.9.1.2 Assumptions and Data Sources

(none)

5.9.1.3 Equations

(none)

5.9.2 Table Driven Command and Control Submodule

5.9.2.1 Operation

The Table Driven Command and Control Submodule was adopted virtually intact from the old MAFIA V simulator which was adapted as the basis for CATTS. The modifications made primarily involve increasing maximum table sizes. There are also modifications to prevent table-driven maneuver decisions for any unit currently under event-driven maneuver control (described in Section 5.9.1). Since the submodule described in this section (5.9.2) has been little changed, the interested reader is strongly recommended to consult the MAFIA V User's Manual, which contains detailed descriptions of the data tables and decision criteria used. The MAFIA V Programming Report can also be a valuable source of information, as it shows detailed logic flows for each subroutine. CATTS Data Requests 13 and 17 also contain seventy-eight pages of explanations of value to someone trying to understand the CATTS Table Driven Command and Control Submodule. Finally, Section 5.9.17 of the CATTS Trainer Programming Report contains twenty pages of description of this submodule.

Subroutine linkages for the Table Driven Command and Control Submodule are shown in Figure 5-146. Table 5-62 gives a brief description of each subroutine, along with its principal inputs and outputs.

There are 3 data base tables that control the table driven command and control submodule. These are the Change of State Table (IOPTB1, IOPTB2), the Movement Code Change Table (MVCHG1, MVCHG2), and the Op State Updating Table (MVCHG3). These 3 tables were the old MAFIA V command and control method. This command and control includes control over maneuvering and firing and some effect on tasking organization. The table command and control provides control by changing the following items: operational state (IOPSTU(I)), movement code (MVTCD(I) & MTCDOG(IOG)), travel code (ITRAV(I), ITRAVG(IOG)), deployment code (IDPCOD(IOG)), and leaving or

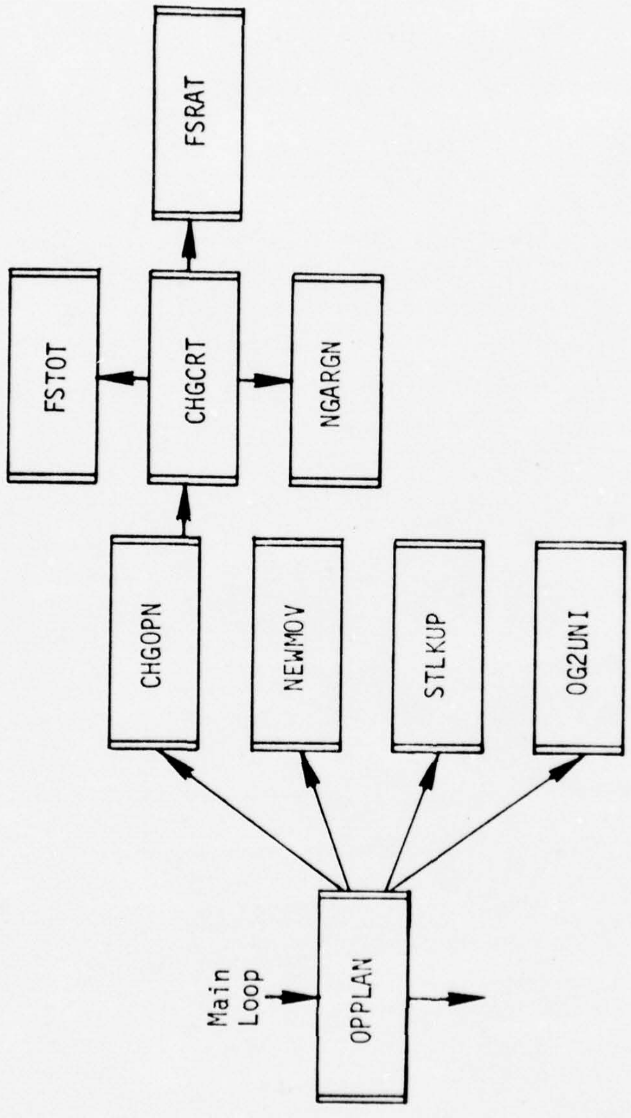


Figure 5-146. Table-Driven Command and Control Subroutine Linkage

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description

Subroutine	Description	Principal Inputs/Outputs		
CHGCRT	Called by subroutine CHGOPN to determine whether a given unit has satisfied a given change of state criterion. See the discussion in the text.	Input:	IDUM	- Number of unit for which calculations are to be performed.
			NTEST	- Number of change of state table entry in use.
			DECVAL(NTEST)	- Input threshold value to use with change of state table entry NTEST.
			NCRIT	- Number of change od state criterion specified in change of state table entry NTEST. See Table 5- .
			IDELAY(IU)	- Delay counter for unit IU, in timesteps.
		Output:	IDELAY	- (See Input).
			NEWS	- New operational state IOPSTU to which unit IU should now be assigned.
CHGOPN	Called by subroutine OPPLAN to determine whether a given unit has satisfied conditions specified in the change of state table to make it eligible for a change of operational state.	Input:	IDUM	- Number of unit for which calculations are to be performed.
			IOPTB1(I,J)	- Change of state selection table. Each entry I is composed of two packed words (J = 1, 2) identifying a situation to be matched. See the detailed discussion in Section 5.9.2.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
CHGOPN (Cont'd)		Input: IOPTB2(I) - Change of state results table. Each entry I is a packed word which identifies action to be taken when a match with IOPTB1 (I,1) and IOPTB1(I,2) occurs. See detailed discussion in Section 5.9.2. NEWS - (See subroutine CHGCRT). Output: IDPCOP(J) - Deployment code of operational group J. 0 = Not deployed, cannot move. 1 = Not deployed, can move. 2 = Technically deployed. 3 = Actually deployed. INOG - (See subroutine AIREVENT). IOPSTU - (See subroutine AIREVENT). MTCDOG(J) - Movement code of operational group J. Corresponds to unit variable NVTCD. See Section 5.5 for a full explanation. MVTCD(IU) - Movement code of unit IU. See Section 5.5 for a full explanation. NCRIT - Change of state criterion number specified by table IOPTB2(NTEST).

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
CHGOPN (Cont'd)		Output: NTEST - Index of entry in IOPTB1 for which a match was found; hence entry of IOPTB2 and DECVAL to use.
FSRAT	Called by subroutine CHGCRT to calculate the firepower contributed to support fire to the total firepower ratio.	Input: ICOL - 1 if calculations are for red forces, 2 if for blue. FPSUM(I,J) - Firepower numbers for red (J = 1) and blue (J = 2) forces. I = 1 Sum of number of indirect and support fire weapons manned (from USEQU) times importance as a target (from UBE). I = 2 Sum of number of support fire rounds which fell on units this timestep (from SWFU). I = 3 Same as I = 2, except some units are excluded from summation based on complicated geometric considerations. I = 4 FPSUM(4,J) = $\frac{\text{FPSUM(1,J) FPSUM(3,K)}}{\text{FPSUM(2,K)}}$ where K = 3 - J. (That is, K represents the other force not represented by J).

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs		
FSRAT (Cont'd)		Output:	FPSUM	- (See Input). FPSUM(4,J) is calculated by FSRAT.
			XTN	- Sum of XTN input value and FPSUM(4,ICOL).
			XTD	- Sum of XTD input value and FPSUM(4,3-ICOL).
FSTOT	Called by subroutine CHGCRT to calculate firepower scores for a given unit.	Input:	II	- Number of unit for which calculations are to be performed.
			ICOL	- 1 if unit II is red, 2 if blue.
			IEQCOD	- (See subroutine AIREVENT).
			USEEQU(IU,K)	- Number of pieces of the Kth equipment type belonging to unit IU which are currently warned. Table *ITEQU(IU,K)* gives the equipment type number of this equipment.
			ITEQU	- (See subroutine AIREVENT in Section 5.9.2).
			IOCOD	- Set of flags used to control operation of various subroutines.
			IOCOD(3)	- Force ratio control.
				0 = No support fire.
				1 = Include support fire.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
FSTOT (Cont'd)		Input: IOCOD(16) - Force ratio control. 0 = Include only support fire weapons. 1 = Include indirect fire weapons. UBE(IEQ) - Target value weighting factor for equipment type IEQ. Output: FPSUM(1,ICOL) - Sum of number of pieces of support fire equipment manned (USEEQU) times target value (UBE).
NEWMOV	Called by both subroutine ARRIVE and subroutine OPPLAN to calculate new movement codes for ground units. Calculations made primarily on the basis of input decision tables.	Input: IDUM - Number of unit for which calculations are to be made. KS - Flag indicating which routine called NEWMOV. = 0 Called by ARRIVE = 1 Called by OPPLAN. The distinction is that if called by OPPLAN, no further change in movement code MVTCD is allowed. MVCHG1(I,J) - Ith entry in movement change look-up table. J = 1 Unit type, operational state, and side.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
NEWMOV (Cont'd)		Input: MVCHG1(I,J) (Cont'd) J = 2 Current movement code, operational grouping, and unit number. This table is described in detail in the text. MVCHG2(I) - Movement change results table. When a match is found with MVCHG1(I,1) and MVCHG1(I,2), then MVCHG2(I) gives the new values for movement code MVTCD (or MTCDOG), travel code, ITRAV, deployment code IDPCOD, operational state IOPSTU, and a pointer to the MVDATA table which provides new values for MVDT1, MVDT2, and MVDT3. This table is described in detail in the text. MVDATA(K) - Table of movement data values for use when a unit changes movement code because of the MVCHG1, MVCHG2 tables. Each MVCHG2 entry, when unpacked, provides the index K to access MVDATA for the MVDT1, MVDT2, and MVDT3 values to correspond to the new movement code MVTCD.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
NEWMOV (Cont'd)		Output: MTCDOG, MVDTA1, MVDTA2, MVDTA3, MVTCD, MVDT1, MVDT2, MVDT3 - These variables control the movement of ground units and operational groupings. They are described in detail in Section 5.5.
OPPLAN	Implements table-driven changes to operational state, and accompanying changes to movement code (through calls to CHGOPN and NEWMOV, respectively). Also initializes a number of counters for each unit.	Input: IOPSTU - (See subroutine AIREVENT, Section 5.9.2). MOVECC - (See subroutine MANEUVER, Section 5.9.2). IOBSTATU(IU) - Obstacle status of unit IU. = 0 Unit not stopped at obstacle. = 1 Unit stopped and engineering support available. = 2 Unit traversing obstacle. = 3 Unit stopped, waiting for engineering support. = 4 Unit stopped, needs bridge, but has no engineering support. = 5 Same as 4 except has engineering support. = 6 Unit stopped to prepare for bridge crossing.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
OPPLAN (Cont'd)		Output: DPERS(IU) - Personnel casualties for unit IU during this timestep. DPERSU(IU) - Personnel casualties for unit IU during this timestep if suppression were ignored. SWFU(IU) - Number of rounds of area support fire falling on Unit IU this timestep. IFIRRL(IU,J) - Flags indicating types of fire received (J = 1) and fired (J = 2) by unit IU this time-step. = 0 Both direct and indirect fire. = 1 Indirect fire only. = 2 Direct fire only. = 3 No fire.
CLSPTG	The subroutine CLSPTG is used to identify all enemy units that are a direct threat to friendly forces in engagements (close-support targets). It also identifies enemy units that are near and might become involved in an engagement (interdictory-fire targets). Partial target weights are computed for these targets. Enemy units that are too close to friendly units to be fired at are identified. Both red and blue are examined in this manner.	Input: DIREAX(I,J) - Direction of the I-th engagement axis, blue to red (stored as sin (J=1), cos (J=2)). IBFRNT(I) - Half-frontage of blue force in the I-th engagement. IFEBAB(I,J) - The X (J=1) and Y (J=2) coordinates of center point of blue FEBA in the I-th engagement.

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
CLSPTG (Cont'd)		IFEBAR(I,J) - The X (J=1) and Y (J=2) coordinates of center point of red FEBA (location of controlling red OG or intersection of red FEBA with I-th engagement axis). IRFRNT(I) - Half-frontage of red force in the I-th engagement. IXPHIB(I) - Position of blue FEBA in the direction of blue movement of the I-th engagement. IXPHIR(I) - Position of red FEBA in the direction of red movement of the I-th engagement. POSFAC(I,J,K) If a target unit of type I and color J (I=red, 2=blue) is in special-threat category, and a weapon in category K (=KFICTR) is to be allocated, then its target weight is multiplied by POSFAC (I,J,K). TYPFAC(I,J) - Weighting factor for unit type I, red (J=1) or blue (J=2), that expresses importance of unit as a support fire target. Output: ITCODE(I) - Target marker code indicatir primary (=0) or secondary (=1) targets in UA(I) for unit I. IWT(JWTS) - Target allocation vector for unit JWTS. In a given weapon-target set, if unit

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs		
CLSPTG (Cont'd)				is a weapon unit, IWT contains the number of enemy units at which it can fire. If unit is a target unit, IWT contains the number of enemy units which can fire at it. ($1 \leq JWTS \leq 100$)
			KSAREA(I,J)	- Area of close-support target region in the I-th engagement; red firing against blue region (J=1), blue firing against red region (J=2).
			POSWGT	- Factor to multiply target value by to reflect position advantage.
STLKUP	Searches the MVCHG3 table for a unit which just had an automatic (hard-coded) change of movement code. The result is a possible (if a match is found) change of operational state.	Input:	IDUM	- Unit number (1-100) of unit which is used in the table search.
			MVCHG3	- Operational state updating table which contains code words specifying changes in unit state when unit movement code has been changed by some means other than through the use of MVCHG1. Code words are of the form: TTUUVWWXXZZ where MVCHG3(I,1) = TTUUV MVCHG3(I,2) = WWXXZZ

Table 5-62. Table-Driven Command and Control Submodule Subroutine Description (Continued)

Subroutine	Description	Principal Inputs/Outputs
STLKUP (Cont'd)		$(1 \le I \le 30)$ Such that TT = Unit number UU = Unit type V = 1 or 2 (red or blue) WW = Operational grouping number of unit XX = Movement code of unit (recently changed) ZZ = New operational state of unit 0 in any position means equality of look-up. If WW = 99, any unit not in any operational group, $(1 \le J \le 2)$ Output: IOPSTU(IU) - Operational state of unit IU

not leaving an op group (INOG(I)). The table driven command and control can compliment the menu/events command and control discussed in the previous section, but it may also interfere with the menu/events command and control. Thus, the user must take care when adding new entries to any of the 3 command and control tables. Obviously, new entries should be tested before being used in a game and removed if they do interfere in an undesirable way with the menu/events command and control. There are 2 cases where no interference can occur: 1) any unit under event maneuver command and control cannot be affected by the table driven command and control, and 2) any unit delayed by an obstacle cannot be affected by the table driven command and control. This means that units under menu/event fire control can be interfered with via table driven command and control, and that tasking organization can be affected by the table driven command and control anytime.

There are two parts to each of these 3 tables: a selection portion and an action portion. The selection portion is examined to determine if a change should occur, and if a change is to occur, the numbers in the action portion are put in the appropriate array to make the change occur.

The Change of State Table is the main action table, the driving command and control table. The other 2 are reaction tables that react to changes that occur. The Change of State Table is searched automatically for every unit every time step and only units under maneuver event control (MOVECC(IU) $\neq$ 0) and units delayed by an obstacle (IOBSTATU(IU) = 2) are not compared to the selection word of this table. The Movement Code Change Table is searched only when a unit arrives at a destination or a unit has had its op state changed by the Change of State Table. The Op State Updating Table is only searched after a unit's movement code has been changed by a value set in the program code (many of these automatic movement code changes are in the engagement logic) or after a unit has it's movement code changed to match that of its operational group.

There is also a data array associated with the Change of State Table and another associated with the Movement Code Change Table. The Decision Value array (DECVAL) contains values used with the selection portion of the Change of State Table. The Movement Data array (MVDATA) contains values to be used with the action portion of the Movement Code Change Table.

Descriptions of the specific operations of each of the 3 command and control tables follow.

Change of State Table

The purpose of this table is to serve as the prime command and control table. It changes directly only Operational State (IOPSTU(IU)), and the leaving of an op group by a unit (INOG(IU)). However, a change of state by this table causes the Movement Code Change Table to be searched, and this can change movement code (MVTCD(IU)), travel code of a unit (ITRAV(IU)), deployment code of an op group (IDPCOD(IOG)), op state (IOPSTU(IU)), and movement data (MVDT1(IU) , MVDT2(IU), and MVDT3(IU)). Every time step, the first 2 words (IOPTB1(I,1), IOPTB1(I,2) for each entry number I in the Change of State Table are searched to see if the values match the current values for each unit IU that is active (ISTATU(IU) ≠ 1), and not under menu/event maneuver control (MOVECC(IU) = 0) and not delayed as a result of an obstacle encounter (IOBSTATU(IU) ≠ 2). Each of the entry words (IOPTB1(I,1), IOPTB(I,2) contain an 11 digit number of the form UUVVWXXYYZZ where

UU = unit type (01-20) (00 => any type)

VV = op group number of the unit (01-20) (00 => disregard this factor, 99 => any unit not in an op group)

W = color (1 = red, 2 = blue, 0 = either)

XX = unit's movement code (01-16), (00 => any code, 88 => any engaged code (1-4), 99 => any non-engaged code (5-16))

YY = unit's current op state (01-98) (00 => any state) also, any op state entry evenly divisible by 10 (i.e., 10, 20, 30, etc.) will match any unit op state with the same digit in the tens place. Thus, YY = 20 would match op states 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29. Thus, when putting op states in the mode table, op states which are divisible evenly by 10 should be avoided, because they can't be singled out by this command and control table.

ZZ = unit number (01-99) (00 = any unit number)

If the current values of a unit match all of the data specified in the first 2 words of entry I, the first part of the third word of entry I (IOPTB2(I)) is consulted to see if the change contained in the second part will be made.

If no match for a unit is found, the table search continues with the next unit. This third associated word IOPTB2(I) has the form AABBBCCD where

AA = number of the change-of-state criteria to use (1 through 85 tests performed by the software subroutine CHGCRIT).

BBB = entry number of the decision value in the DECVAL data array to be used with the criterion. If two values are required with a criterion, they must be listed consecutively in the DECVAL array. In this case, the AA above points to the first value, and the program automatically uses the next entry for the second value.

CC = new op state of unit if criterion is satisfied.

D = leave or don't leave current op group (0 => don't leave, >0 => leave).

The criterion number specified in the AA portion of the word is evaluated using the value(s) specified in DECVAL (BBB), where BBB is a 3-digit pointer into the DECVAL array (if more than one value is required, BBB points to the first). If the test is passed, the operational state of the unit is set equal to the value specified in item CC of the IOPTB2 word. Finally, code D indicates whether or not the unit should leave its operational grouping.

If the test is not passed, the table search for the unit continues.

After a unit undergoes a change in its operational state, a second pass is made through the IOPTB1 array to determine if a second change is required (the new op state is used in the second search). No more than two operational state changes will be made to a unit in a given time step. If a change of state occurs, the Movement Code Change Table is searched to determine whether the movement code should also be changed. Even though the Movement Code Change Table can change op state, when searched as a result of an op state change, the Movement Code Change Table is not allowed to make op state changes.

If the value of IOPTB2, representing criteria number, etc., is negative, it is set to positive before extracting the packed data. After processing the data, the corresponding entry in IOPTB1(i,1) is set to 999999999, effectively eliminating that entry from the table for the remainder of the exercise.

There are 85 different criteria evaluated in subroutine CHGCRT. These criteria are hard-coded in the subroutine. A list of the 85 criteria can be found in Table 5-63, where the unit being processed is referred to as "unit I," and the threshold value is "IDATA" (second threshold value, if applicable, is "IDATA1").

Movement Code Change Table

The purpose of this table is to react to 2 circumstances:

- units arriving at destination
- units changing op states as a result of Change of State Table (COS) entries

It directly changes movement code (MVTCD(IU)), travel code (ITRAV(IU)), deployment code (IDPCOD(IOG)), op state (IOPSTU(IU)), and movement data (MVDT1(IU), MVDT(IU), MVDT3(IU)). When a unit arrives or gets a COS op state change, the first 2 words (MVCHG1(I,J),J = 1, 2) of the Movement Code Change Table are compared to the units' current values. If all of the values match, the appropriate unit variables are set equal to the values in the associated third word (MVCHG2(I)) of entry I, and the movement data variables are set equal to the values from the movement data array (MVDATA(J)) (pointed to by MVCHG2).

The first 2 words (MVCHG1(I,J),J = 1, 2) of each entry have the form UUVVWXXYYZZ where

UU = unit type (01-20) (00 => any type)

VV = op state of unit (01-98) (00 => any state), also any op state entry evenly divisible by 10 (i.e., 10, 20, 30, etc.) will match any unit op state with the same digit in the ten's place. Thus, VV = 20 would match current unit op states 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29. Thus, when putting op states in the mode table, op states which are evenly divisible by 10 should be avoided, because they can't be singled out by this command and control table.

W = color (1 = red, 2 = blue, 0 = either)

XX = movement code of unit (01-16) (00 => any code)

YY = op group number (01-20) (00 => disregard this factor, 99 = any unit not in an op group)

ZZ = unit number (01-99) (00 => any unit)

Table 5-63. Change of State Criteria

(1) time $\geq$ IDATA

(2) time = IDATA

(3) x coordinate of unit I $\geq$ IDATA

(4) x coordinate of unit I $<$ IDATA

(5) x coordinate of unit I (rounded to nearest 100) = IDATA

(6)* Distance from unit I to nearest enemy unit <u>in same engagement</u> $\leq$ IDATA

(7)* Distance from unit I to nearest enemy unit <u>in same engagement</u> $>$ IDATA

(8) Distance from unit I to nearest enemy unit $\leq$ IDATA

(9) Distance from unit I to nearest enemy unit $>$ IDATA

(10)* Distance between unit I and enemy FEBA $\leq$ IDATA

(11)* Distance between unit I and enemy FEBA $>$ IDATA

(12)* Distance between unit I and friendly FEBA $\leq$ IDATA

(13)* Distance between unit I and friendly FEBA $>$ IDATA

(14)* x coordinate of the location of the enemy FEBA $\leq$ IDATA

(15)* x coordinate of the location of the enemy FEBA $>$ IDATA

(16)* x coordinate of the location of the friendly FEBA $\leq$ IDATA

(17)* x coordinate of the location of the friendly FEBA $>$ IDATA

(18)* Distance between the opposing FEBAs $\leq$ IDATA

(19)* Distance between the opposing FEBAs $>$ IDATA

(20) Degree of suppression of unit I $\geq$ IDATA

(21) Degree of suppression of unit I $<$ IDATA

(22) Not used

(23) Unit I is below critical minimum level for at least one principal ammunition type

(24) Unit I is above critical minimum level for at least one principal ammunition type

(25) Unit I is below critical minimum level for all principal ammunition types

(26) Unit I is above critical minimum level for all principal ammunition types

(27) $1 - \frac{m_I(t)}{m_I(0)} \geq$ IDATA (fraction of men lost in unit I)***

Table 5-63. Change of State Criteria (Cont'd)

(28) $1 - \frac{n_I 1(t)}{n_I 1(0)} \geq$ IDATA (fraction of first equipment type in unit I that is lost) ***

(29)* $1 - \frac{\Sigma m_J(t)}{\Sigma m_J(0)} \geq$ IDATA (for friendly units J in the same engagement as unit I with a status code equal to 1) ***

(30) $\frac{1}{m_I(0)} \times \frac{\delta m_I}{\delta t} \geq$ IDATA ***

(31) $\frac{1}{m_I(0)} \times \frac{\delta m_I}{\delta t} <$ IDATA ***

(32) $\frac{1}{n_{I_1}(0)} \times \frac{\delta n_{I_1}}{\delta t} \geq$ IDATA ***

(33) $\frac{1}{n_{I_1}(0)} \times \frac{\delta n_{I_1}}{\delta t} <$ IDATA ***

(34)* $\frac{1}{\Sigma m_J(0)} \times \frac{\Sigma \delta m_J}{\delta t} \geq$ IDATA (for friendly units J in the same engagement as unit I with a status code equal to 1) ***

(35)* $\frac{m_I(0) - m_I(t)}{m_I(0)} - \frac{\text{distance to enemy FEBA}}{\text{unit I movement rate} * m_I(0)} \frac{\delta m_I}{\delta t} \geq$ IDATA ***

(36)* The Force Ratio** $\geq$ IDATA (for all friendly units J and enemy units K in the same engagement as unit I with a status code of 1)

(37)* The same as 36 except that all units J and K must be on the same side of the Engagement Axis as unit I

(38)* The Force Ratio** $<$ IDATA (for all friendly units J and enemy units K in the same engagement as unit I with a status code of 1)

(39)* The same as 38 except that all units J and K must be on the same side of the Engagement Axis as unit I

Table 5-63. Change of State Criteria (Cont'd)

(40)* Position of unit IDATA $\leq$ IDATA1 from its enemy FEBA

(41)* Position of operational grouping IDATA $\leq$ IDATA1 from its enemy FEBA

(42)* Position of unit IDATA $>$ IDATA1 from its enemy FEBA

(43)* Position of operational grouping IDATA $>$ IDATA1 from its enemy FEBA

(44) Unit IDATA is in state IDATA1

(45) The forward-most unit of operational grouping IDATA is in state IDATA1

(46)* For the engagement of unit IDATA, the distance between the Red and Blue FEBAs $\leq$ IDATA1

(47)* For the engagement of unit IDATA, the distance between the Red and Blue FEBAs $>$ IDATA1

(48)* For the engagement of operational grouping IDATA, the distance between the Red and Blue FEBAs $\leq$ IDATA1

(49)* For the engagement of operational grouping IDATA, the distance between the Red and Blue FEBAs $>$ IDATA1

(50) Unit IDATA is engaged

(51) Operational grouping IDATA is engaged

(52) Unit IDATA is not engaged

(53) Operational grouping IDATA is not engaged

(54)* For the engagement of unit IDATA, the Force Ratio** in the Close Support Region $<$ IDATA1

(55)* For the engagement of unit IDATA, the Force Ratio** in the Close Support Region $\geq$ IDATA1

(56)* For the engagement of unit IDATA, the Force Ratio** in the Close Support and Interdiction Regions $<$ IDATA1

(57)* For the engagement of unit IDATA, the Force Ratio** in the Close Support and Interdiction Regions $\geq$ IDATA1

(58)* Same as 54 except use operational grouping IDATA

(59)* Same as 55 except use operational grouping IDATA

Table 5-63. Change of State Criteria (Cont'd)

(60)* Same as 56 except use operational grouping IDATA

(61)* Same as 57 except use operational grouping IDATA

(62)* For the engagement of unit IDATA, the Force Ratio** in the left half-sector of the Close Support Region $<$ IDATA1

(63)* For the engagement of unit IDATA, the Force Ratio** in the left half-sector of the Close Support Region $\geq$ IDATA1

(64)* For the engagement of unit IDATA, the Force Ratio** in the right half-sector of the Close Support Region $<$ IDATA1

(65)* For the engagement of unit IDATA, the Force Ratio** in the right half-sector of the Close Support Region $\geq$ IDATA1

(66)* Same as 62 except operational grouping IDATA

(67)* Same as 63 except operational grouping IDATA

(68)* Same as 64 except operational grouping IDATA

(69)* Same as 65 except operational grouping IDATA

(70) Destination unit or operational grouping is defunct

(71) Destination engagement is defunct

(72) Rounds of support weapon fire/unit area falling on unit I $\geq$ IDATA

(73) Rounds of support weapon fire/unit area falling on unit I $<$ IDATA

(74) Status code of unit IDATA = 1

(75) Status code of unit IDATA $\neq$ 1

(76) x-coordinate of unit IDATA $\leq$ IDATA1

(77) x-coordinate of unit IDATA $>$ IDATA1

(78) x-coordinate of operational grouping IDATA $\leq$ IDATA1

(79) x-coordinate of operational grouping IDATA $>$ IDATA1

(80) This criterion sets the delay counter to IDATA unless the delay counter is not zero; normally, no change-of-state should be indicated

(81) This criterion decrements the delay counter of the unit by the magnitude of the time increment. A change-of-state will occur when the delay counter is less than or equal to the time increment

Table 5-63. Change of State Criteria (Cont'd)

(82) Difference between x-coordinates of units I and IDATA $\leq$ IDATA1

(83) Difference between x-coordinates of units I and IDATA > IDATA1.

(84) (Distance between units I and IDATA) $\leq$ IDATA1.

(85) (Distance between units I and IDATA) > IDATA1.

*Unit I must be engaged or the test automatically fails.

**The Force Ratio is $\dfrac{\sum_{J\beta}\sum u_{\beta} n_{J\beta} + \text{Measure of friendly support fire}}{\sum_{Kb}\sum u_{b} n_{Kb} + \text{Measure of enemy support fire}}$

where β ranges over all equipment types in the friendly unit J.

b ranges over all equipment types in the enemy unit K.

u is the equipment <u>weight</u> for equipment number <u>n</u>

***Variables used in defining criteria 27 through 35 are defined as follows:

$m_I(0)$ = number of personnel in unit I at time 0.

$m_I(t)$ = number of personnel in unit I at time t.

$\dfrac{\delta m_I}{\delta t}$ = number of men lost in last time step.

$n_{I_1}(0)$ = number of pieces of first equipment on unit I's equipment list at time 0.

$n_{I_1}(t)$ = number of pieces of first equipment on unit I's equipment list at time t.

$\dfrac{\delta n_{I_1}}{\delta t}$ = number of pieces of first equipment on unit I's equipment list in last time step.

The action word - the third word (MVCHG2(I)) - has the form EEFGJJKKK, where

EE	=	new movement code of unit
F	=	new travel code of unit (0 => no change)
G	=	new deployment code (only a value of 1 has any effect)
JJ	=	new op state of unit (00 => no change. 00 must be used with entries that are to be activated due to COS op state change)
KKK	=	entry number of movement data (MVDATA) of the first movement data associated with the new movement code. Up to three values can be needed, and these must be consecutive in the MVDATA array. See table of movement codes for which of the 3 values are required.

If no match occurs with the MVCHG1 words, the unit will keep it's current movement values.

Note that finding a match among the entries of array MVCHG1 does not guarantee a movement change. A very important restriction is the following: units may not be changed from an unengaged (5-16) to an engaged (1-4) movement code. Such a change is performed automatically by built-in software during arrival and engagement processing. The same is true when a unit's movement code is being changed to 4 or 16 (deployed and waiting). Built-in software will handle this, and any attempt to produce such a change (via arrays MVCHG1 and MVCHG2) is ignored. Also, if the movement code change is the result of an op state change from the Change of State Table, is a new state (JJ ≠ 00) is specified, the whole action word (MVCHG2) will be ignored. Also, if an arriving unit belongs to an op group, and another unit in the op group has already arrived and gotten the movement code for itself and the op group, the table will not be searched for the unit.

Op State Updating Table

The purpose of this table is to change a unit's op state after it's movement code has been changed by some means other than the Movement Code Change Table or the menu events. Basically, this allows an appropriate op state change for a unit that gets a hard coded movement code. This table changes only the op state (IOPSTU(IU)) of a unit.

A unit can have its movement code changed automatically by the engagement logic. Also, a unit can have its movement code changed to match that of the op group to which it belongs. After all such instances of movement code change, the current values of a unit are matched against the 2 words of the I entries in the Op State Updating Table (MVCHG3(I,J), J = 1, 2). These words have the form TTUUVWWXXZZ, where

TT = unit number (01-99) (00 => any number)

UU = unit type (01-20) (00 => any type)

V = color (1 = red, 2 = blue, 0 = either)

WW = op group number of the group the unit belongs to (01-20) (00 => disregard this factor, 99 => any unit not in an op group)

XX = movement code of unit - recently changed (01-16) (00 => any code)

ZZ = new op state of unit

If the unit matches all the values of one of the entries in this table, the units op state (IOPSTU(IU)) will be set equal to the value specified in the ZZ portion of the matching entry. Thus, units that receive hard coded movement code changes can be made to change to appropriate op states.

5.9.2.2 Assumptions and Data Sources

There are some hard coded command and control decision rules relevant to this submodule. These involve changing a unit's movement code (MVTCD(IU) and movement data (MVDT1(IU), MVDT2(IU), and MVDT3(IU)) by subroutines NEWMOV and ARRIVE. All of these hard coded rules are detailed in the flow diagrams of these 2 subroutines in Section 5.5.2.

These assumptions were made by the original MAFIA programmers.

5.9.2.3 Equations

(None)

5.10 MISCELLANEOUS AND ANCILLARY MODULE

The subroutine and function subprograms described in this section perform support calculations generally used in more complex computations in other modules. These are primarily interpolations, geometric calculations, and utility functions. Most of the routines are written in FORTRAN and taken, in most cases, from the MAFIA V Program. Xerox assembly language has been used in cases where it proved to increase computational efficiency.

With the exception of the two subroutines described in Sections 5.10.30 and 5.10.34, REDIST and USEFUEL, respectively, all other subroutines described herein are designed to solve fairly simple geometric problems or to do table look-up operations.

5.10.1 Subroutine CHKLOS(IX1, IX2, IX,K)

5.10.1.1 Operation

This subroutine determines whether or not IX has a value between IX1 and IX2. If IX is less than IX2 and greater than IX1, or less than IX1 and greater than IX2, K is set to 2. Otherwise, K is set to 1. IX1, IX2, and IX are inputs to this subroutine. K is output by this subroutine.

5.10.1.2 Assumptions and Data Sources

(None)

5.10.1.3 Equations

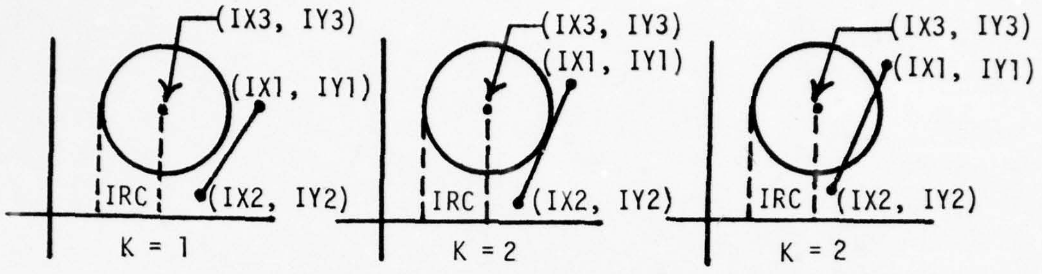
$IX1 < IX < IX2$

$IX2 < IX < IX1$

5.10.2 Subroutine CKLOSC(IX1, IY1, IX2, IY2, IRC, IX3, IY3, K)

5.10.2.1 Operation

This subroutine is used to determine whether the line segment defined by (IX1, IY1) (IX2, IY2) intersects a circle of radius IRC centered at (IX3, IY3) If an intersection exists, K is set to 2. If no intersection exists, K is set to 1.



IX1, IY1, IX2, IY2, IX3, IY3, IRC are inputs to this subroutine. K is output by this subroutine.

5.10.2.2 Assumptions and Data Sources

(None)

5.10.2.3 Equations

Compute distance DIST from the center of the circle to the point on the line closest to the center of the circle.

$$DIST = ((X3 - XX)^2 + (Y3 - YY)^2)^{1/2}$$

where

(X3, Y3) are (X, Y) coordinates of circle center

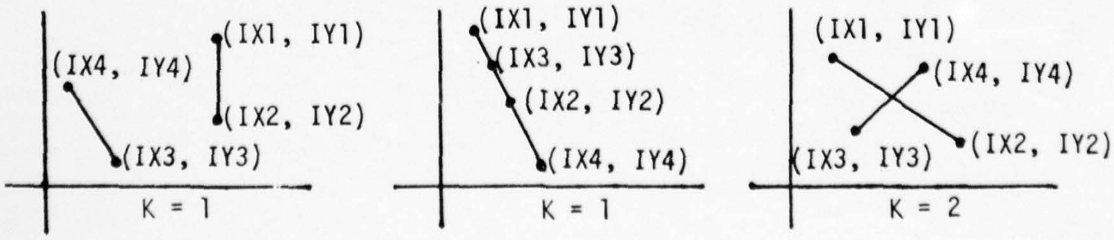
(XX, YY) is the point on the line closest to circle center

5.10.3 Subroutine CKLOSL(IX1, IY1, IX2, IY2, IX3, IY3, IX4, IY4, K)

5.10.3.1 Operation

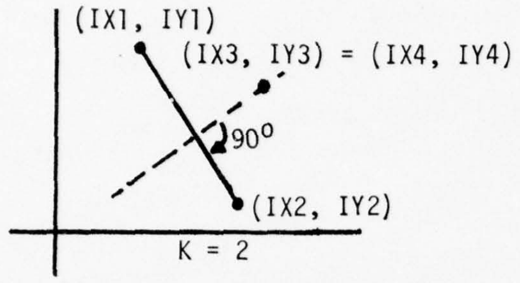
This subroutine determine whether two line segments, defined by (IX1, IY1),(IX2, IY2) and (IX3, IY3), (IX4, IY4), intersect.

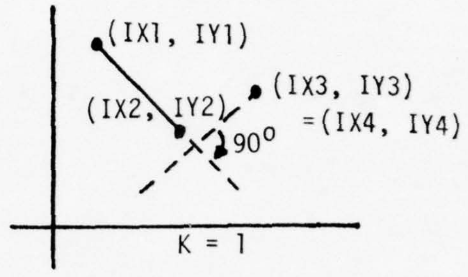
If an intersection exists, K is set to 2; otherwise, K is set to 1. Parallel line segments are not conisdered to intersect even if they are identical.



Other special cases are handled as follows:

a) If either line segment, but not both, defines a single point, the projection of the point on the line segment is used as the intersection point.





b) If both line segments are merely single points, K = 1 is returned unless the two points coincide.

c) Intersections that occur on end points of line segments are considered to be valid unless the two line segments are parallel.

IX1, IY1, IX2, IY2, IX3, IY3, IX4, IY4 are inputs to this routine, K is output by this routine.

5.10.3.2 Assumptions and Data Sources

It is assumed that no intersections will occur with X or Y values greater than the magnitude of ±7999999.

5.10.3.3 Equations

(None)

5.10.4 Subroutine CKLOS2(IX1, IY1, IX2, IY2, IX3, IY3, IX4, IY4, IX, IY, K)

5.10.4.1 Operation

This subroutine computes the coordinates of the point of intersection of two line segments, defined by (IX1, IY1), (IX2, IY2) and (IX3, IY3), (IX4, IY4). It returns a flag (K) indicating whether or not an intersection exists. IX and IY are returned with the coordinates of that intersection. K is returned with value 2 if the intersection calculated lies within the X coordinates of the two segments; otherwise, K is returned with value 1. All special cases are handled as with subroutine CKLOSL (see Section 5.10.3), except that if exactly one of the segments is parallel to the X axis, a K value of 2 may be returned even if the segments do not intersect.

IX1, IY1, IX2, IY2, IX3, IY3, IX4, IY4 are inputs to this routine and IX, IY, and K are output.

5.10.4.2 Assumptions and Data Sources

Same as subroutine CKLOSL (see Section 5.10.3.2).

5.10.4.3 Equations

$IX1 \le IX \le IX2$

$IX3 \le IX \le IX4$

5.10.5 Subroutine CK4XING

5.10.5.1 Operation

Subroutine CK4XING has two main purposes. The principal of these is to generate an alert message whenever a control measure is "violated". The secondary is to help the support fire allocation logic in subroutine SPTALO to avoid automatic support fire allocation to targets which would violate firing control measures. CATTS as delivered to Fort Leavenworth had automatic support fire allocation disabled through data base inputs; therefore, this secondary purpose has never been thoroughly tested and has never been used in an exercise.

Control measures in CATTS are identified by two "type" numbers. One of these is just a straightforward listing of types with associated numbers. This is called the "foreground" control measure type, as it is used primarily by the foreground display software. The other "type" is obtained from table MTYPOFCM by subroutine CONTMS at the time the measure is created, and is called the "background" type, as it is used by subroutine CK4XING. Both numbers are stored in the control measure table ICM. The background "type" is similar to the foreground "type" except that unused type numbers have been compressed out to save storage. Also, the background "type" is set to zero for all measures for which no alert messages are to be generated (such as maneuver routes). Further, a distinction must be made between control measures for movement and for firing. Therefore, all firing control measures have 64 added to their "type" number. But, some firing control measures may not be fired short of (in other words, must be fired across). In order to distinguish between these, those measures which must be fired across have an additional 64 (or total of 128) added to their type numbers.

Each control measure in CATTS must also belong to a "unit". This unit number is stored in the table ICM. For flexibility, however, "unit" has been expanded to include operational groupings and units not completely modeled (such as blue division, for example).

Each "unit" in CATTS, including op groups and imcompletely modeled ones, is assigned a next higher command unit by the table NXHGCM. A value of zero is possible and indicates a unit for which no higher command is included in the simulation.

CK4XING considers a control measure to apply only to the "unit" to which the measure is assigned at creation or to some "unit" which, by a chain of next higher command values from NXHGCM, is subordinate to that assigned "unit".

The actual processing for CK4XING begins with a call which specifies a unit (IUNIT), a source point (IX1,IY1), a destination point (IX2,IY2), and a flag indicating whether the calling subroutine is concerned with movement (IMFIRING = 0) or firing (IMFIRING = 1).

The first step is the construction of a table NCOMM which lists the complete chain of command for unit IUNIT. The next step is to cycle through each control measure in the model table ICM. If the first word of the measure is zero, it has been deleted and is skipped. If the background control measure "type" as explained above does not agree with the value of IMFIRING, the measure is skipped. If the measure is red and IUNIT is blue, or vice versa, the measure is skipped (a measure can belong to both red <u>and</u> blue, however). If the measure belongs to a "unit" not listed in table NCOMM, the measure is skipped.

Now geometric processing begins. If the minimum or maximum X or Y coordinates (stored in table ICM) of the measure are such that the line segment (IX1, IY1) to (IX2, IY2) could not cross the measure, then the measure is skipped. Otherwise, the line segments defining the control measure are unpacked (even control points are modeled as small areas) and the actual number (ICROSS) of these segments which cross the given segment is calculated via calls to subroutine CKLOSL. For movement measures, alerts are generated where applicable (an odd number of crossings is required to generate an alert, except for control points), and processing is complete for the affected measure. For firing measures, the control measure number is stored in table MSRVIOL if it has been violated. Whether a violation has occurred depends on whether the number of crossings is even or odd and whether this type of measure must or must not be crossed. Processing then ends for this measure.

Subroutine CK4XING terminates processing when all control measures have been considered. However, it has an entry point, CMFIRAL, which is called when firing actually occurs in violation of a measure. CMFIRAL has, as arguments, a unit number and a control measure number (from the MSRVIOL table calculated earlier) and generates the necessary alert message before returning.

5.10.5.2 Assumptions and Data Sources

(None which are not implicit in the operation description.)

5.10.5.3 Equations

(None)

5.10.6 Subroutine CMSEGMNT (IUNIT, K, ISEGMENT, IRETURN, ICNTRLX, ICNTRLY)

5.10.6.1 Operation

This routine finds and unpacks the next destination point for a ground unit traveling on an existing route (existing routes are modeled as control measures).

Input:	IUNIT	= number of unit.
	K	= number of control measure which represents route being traveled by IUNIT.
	ISEGMENT	= number of the line segment on control measure K which IUNIT has just completed traversing.

Output: IRETURN = 0 is the normal case when the next point has been found.

$\left.\begin{array}{l}\text{ICNTRLX}\\ \text{ICNTRLY}\end{array}\right\}$ = (X, Y) coordinates of next destination point (if any).

5.10.6.2 Assumptions and Data Sources

Existing ground movement routes are modeled as special control measures, and stored in table ICM.

5.10.6.3 Equations

(None)

5.10.7 Subroutine DESTIN

5.10.7.1 Operation

DESTIN performs a standard coordinate rotation and translation of input point P = (IX2, IY2). P is first rotated by an angle defined by sine and cosine ST and CT respectively. It is then translated by = (IX1, IY1) to give the resultant point (IX3, IY3). This is performed via calls to ITRANS.

5.10.7.2 Assumptions and Data Sources

(None)

5.10.7.3 Equations

(None)

5.10.8 Subroutine FIRAGEN

5.10.8.1 Operation

Called once per timestep to maintain tables which keep track of which units are firing which types of weapons at which other units. Also maintains tables which give casualty levels since last report. The sole purpose of these tables and of this routine is to determine when alert messages should be generated, and to generate those messages.

The first type of alert message is generated whenever a unit starts firing or receiving fire in any of the following categories: direct fire, indirect fire, support fire. The tables controlling this alert are OLDFLG, which describes the situation last timestep, and NOWFLG, which describes the current timestep. An alert is generated when NOWFLG indicates that a unit is now firing (or receiving) a class of fire which OLDFLG indicates it was not firing (or receiving) last timestep.

The second type of alert message is a casualty report. This message is generated whenever a unit which was under fire receives no fire at all for a full timestep, and reports total men and equipment casualties for the engagement. It is also generated whenever casualties incurred since the last casualty report exceed an input threshold. For equipment type IEQ, input variable IECTH (IEQ) gives the number of pieces of that equipment which, when lost, will be sufficient to generate a casualty alert. Input variable PCTH specifies the fraction of personnel which, when lost, will be sufficient to generate a casualty alert.

The actual functioning of FIRAGEN is straightforward but tedious, since most of the tables involved are packed, and therefore accessed with assembly language.

5.10.8.2 Assumptions and Data Sources

(None)

5.10.8.3 Equations

(None)

5.10.9 Function IRND(X)

5.10.9.1 Operation

This function adds +.5 or -.5 to a floating point number, X, depending on whether the number is positive or negative, respectively, and outputs the results in its integer truncated value.

5.10.9.2 Assumptions and Data Sources

(None)

5.10.9.3 Equations

$IRND = X \pm .5$

5.10.10 Function ISFIROR(IU)

5.10.10.1 Operation

This function searches the fire override table (array IFIROVRD in common block FIROVRD) for a reference to unit number IU. If a reference to IU is found, a 2 is returned. If such a reference is not found, a 1 is returned.

5.10.10.2 Assumptions and Data Sources

It is assumed that entries in IFIROVRD are packed in bytes in ascending order.

5.10.10.3 Equations

(None)

5.10.11 Function ITRANS(K1, X1, K2, X2)

5.10.11.1 Operation

This function performs a calculation frequently used in coordinate transformation. The calculation consists of converting the integerized coordinate point (K1 and K2) to floating point numbers, multiplying each by floating point numbers (X1 and X2, respectively), summing the products and converting the result to integer form by calling IRND.

The standard geometry rotation formulas that this function uses are:

$X^1 = XCOS\theta + YSIN\theta$ or X^1 = ITRANS(IX, COS(θ), IY, SIN(θ))

$Y^1 = YCOS\theta - XSIN\theta$ or Y^1 = ITRANS(IY,COS(θ), - IX, SIN(θ)

$X = X^1COS\theta - Y^1SIN\theta$

$Y = Y^1COS\theta + X^1SIN\theta$

where primed letters designate new coordinates, and θ is the angle of rotation.

5.10.11.2 Assumptions and Data Sources

(None)

5.10.11.3 Equations

ITRANS = K1 * X1 + K2 * X2

5.10.12 Subroutine LATDST(I,K)

5.10.12.1 Operation

This subroutine makes use of subroutine PLDST2 to calculate the perpendicular lateral distance of unit I from the axis of engagement K. The calculated distance added to the unit's half-frontage is used to replace engagement K's half-frontage if it is larger than the existing half-frontage. If the unit is in the process of deploying, the distance is calculated from the deployment point to the engagement axis.

5.10.12.2 Assumptions and Data Sources

(None)

5.10.12.3 Equations

(None)

5.10.13 Subroutine LLINT(A1, B1, A2, B2, IX, IY)

5.10.13.1 Operation

This subroutine computes the coordinates (IX, IY) of the intersection of two lines, which are defined in slope-intercept form: $Y1 = A1 \times X1 + B1$ and $Y2 = A2 \times X2 + B2$. The intersection is obtained by the simultaneous solution, by algebraic means, of the two equations and is then integerized. If the two slopes are equal, IX and IY are set, respectively, equal to -7999999 and +7999999, for a negative slope, and to +7999999 and +7999999, for a positive slope.

5.10.13.2 Assumptions and Data Sources

(None)

5.10.13.3 Equations

$$IX = (B2 - B1)/(A1 - A2)$$

$$IY = A1 * IX + B1$$

5.10.14 Subroutine LLINT1(IX1, IY1, IX2,IY2, IX3, IY3, IX4, IY4, IX, IY)

5.10.14.1 Operation

This subroutine calculates the point of intersection of two lines, each defined by two points.

If either line is really a point (defining points co-located), its slope is set equal to the negative inverse of the other line's slope, thus, projecting the point (line 1) onto the second line. Once the special cases above have been checked for and treated, the first and second lines are both converted to the slope-intercept form, and calculated as in Subroutine LLINT, of Section 5.10.13.

5.10.14.2 Assumptions and Data Sources

It is assumed that no intersections will occur with X or Y values greater than the magnitude of ±7999999.

5.10.14.3 Equations

A1 = FLOAT(IY2 - IY1)/FLOAT(IX2 - IX1)

Line 1 & 2 intercept equations are:

B1 = FLOAT(IY2) - FLOAT(IX2) (A1)

B2 = FLOAT(IY2) - FLOAT(IX3) (A2)

5.10.15 Subroutine LLINT2(IX1, IY1, IX2, IY2, IX3, IY3, A2, IX, IY)

5.10.15.1 Operation

This subroutine computes the coordinates (IX, IY) of the intersection of two lines, the first defined by the two points (IX1, IY1), (IX2, IY2) and the second defined by point (IX3, IY3) and slope A2. This is done by calculating the slope of line one and the Y-intercepts of each line and proceeding as in Subroutine LLINT, of Section 5.10.13.

5.10.15.2 Assumptions and Data Sources

It is assumed that no intersections will occur with X or Y values greater than the magnitude of ±7999999.

5.10.15.3 Equations

The Y intercepts of both lines are calculated via:

B1 = IY1 - IX1 * A1

B2 = IY3 - IX3 * A2

5.10.16 Subroutine LLINT3(IX1, IY1, IX2, IY2, A1, A2, IX, IY)

5.10.16.1 Operation

This subroutine computes the coordinates (IX, IY) of the intersection of two lines, L1 and L2, each defined by one point and slope, IX1, IY1, A1 and IX2, IY2, A2, respectively. The Y-intercepts are first computed for each line. Subroutine LLINT is then caleed to compute the intersection point of the lines from the slope-intercept form, Y = AX + B.

5.10.16.2 Assumptions and Data Sources

(None)

5.10.16.3 Equations

(None)

5.10.17 Subroutine LOADSEG

5.10.17.1 Operation

This subroutine performs 2 functions:

1) It saves the overlay segment number (as the first word of common block JTRAP, i.e., NUMSEG) of the overlay segment about to be loaded into core. This is done so that if a model trap occurs, the user will be able to tell which segment of code was being executed. See "Core Dump Procedure When Math Model Traps" in the Data Base/Operations Manual, page 240.

2) It calls the Xerox library subroutine SEGLOAD(NUMSEG) to load the overlay segment into core.

5.10.17.2 Assumptions and Data Sources

That the user would like to know which overlay segment is executing in core when a trap occurs. TRW engineering estimate by R. Lew.

5.10.17.3 Equations

(None)

5.10.18 Subroutine LOWALERT

5.10.18.1 Operation

Called once per timestep to generate any alert messages necessary to indicate any of the following for non-defunct ground units:

1) Low gasoline level in a unit.

2) Low diesel level in a unit.

3) Significant change in rate of movement for a unit.

4) Unit halted by an obstacle.

5) Casualties caused by minefield.

6) Low ammunition level in a unit.

The three low level alerts (gasoline, diesel, and ammunition) are generated the first timestep that current levels (CLGAS, CLDIES, NETAMU) for a unit fall below input fractions (REVGAS, REVDIES, AMOREV) times initial levels (BLGAS, BLDIES, NETAMX). When a low level alert is generated, a flag is set in a table (IGASF, IDIESF, IAMOALF) to indicate the issuance of the alert. No further alerts of the same kind will be issued as long as the flag is set, and the flag is cleared only when a Resupply event is accepted for the affected unit (whether or not the resupply includes the needed item).

The alert for significant movement rate change occurs whenever the new movement rate (ROMU) for a unit differs from the rate last timestep (BROMU) by either an input threshold amount (DELMNT), or an input threshold fraction (DELMOVE) times the old rate (BROMU). This alert is also generated whenever a unit's movement (from tables MOUNTOLD and MOUNTED) changes from mounted to dismounted, or vice versa, regardless of the rates involved.

Alerts for units stopped by obstacles are generated based on flags set by the obstacle model in tables IOBALRT and IOBSTOP. Alerts for minefield casualties are generated only when an alert has been issued for being stopped by an obstacle which is a minefield. The casualties reported are obtained from tables KLDPPL and NTPEQDST, both set by the obstacle model.

5.10.18.2 Assumptions and Data Sources

(None)

5.10.18.3 Equations

(None)

5.10.19 Function MANNING(IEQ,IU)

5.10.19.1 Operation

This function returns the number of personnel required to operate a given type of equipment in a given unit (IU) as a function of the unit's travel mode. The byte-packed array OPE for equipment IEQ is unpacked to return the crew size required for the operation of the equipment in the mounted or dismounted mode and its troop-carrying capacity. The crew size returned is that for the dismounted mode, unless the equipment is a vehicle in a unit moving in the mounted mode. In this case, the crew size refers to the mounted mode of the equipment.

5.10.19.2 Assumptions and Data Sources

(None)

5.10.19.3 Equations

(None)

5.10.20 Subroutine NEWLOC(IX1, IY1, ST, CT, IDIST, IX2, IY2)

5.10.20.1 Operation

Calculates coordinates of the point (IX2, IY2) reached by traveling a distance IDIST from the point (IX1, IY1) in the direction with sine and cosine equal ST and CT, respectively.

5.10.20.2 Assumptions and Data Sources

(None)

5.10.20.3 Equations

$$IX2 = IX1 + CT \cdot IDIST$$

$$IY2 = IY1 + ST \cdot IDIST$$

5.10.21 Function NXUNIT(IUNIT)

5.10.21.1 Operation

Given a unit, IUNIT, this function returns the unit number of the next unit appearing on the unit list. The number is obtained by unpacking the right-most byte of variable LISTUN(IUNIT).

5.10.21.2 Assumptions and Data Sources

(None)

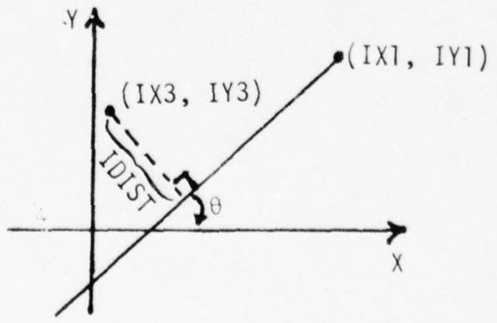
5.10.21.3 Equations

(None)

5.10.22 Subroutine PLDST2(IX1, IY1, ST, CT, IX3, IY3, IDIST)

5.10.22.1 Operation

This subroutine computes the closest distance between a line, L, and a point, P3. Inputs to the subroutine are the description of Line L, defined by a point (IX1, IY1), the sine(ST) and cosine(CT) of the line's positive angle with the X-axis, and the point P3 (IX3, IY3). The computed distance is the output (IIDST). This routine makes use of subroutine PLPRJ2 to compute the closest point on L to P3. If ST and CT are the sine and cosine of angle theta (θ), the value of IDIST is computed as in the following diagram:



5.10.22.2 Assumptions and Data Sources

(None)

5.10.22.3 Equations

Given the closest point (IX2, IY2) on L to P3, the distance between (IX2, IY2) and (IX3, IY3) is calculated via:

$$\text{IDIST} = ((\text{IX3} - \text{IX2})^2 + (\text{IY3} - \text{IY2})^2)^{1/2}$$

5.10.23 Subroutine PLINE(LINE,N)

5.10.23.1 Operation

This I/O routine outputs N lines contained in buffer LINE to the line printer device using the asynchronous foreground line printer control program PRINTQ. The routine calculates successively the starting address of each 136 character line buffer (the first character is carriage control) and passes each line to the foreground PRINTQ program via a constructed CAL4, 1.

5.10.23.2 Assumptions and Data Sources

(None)

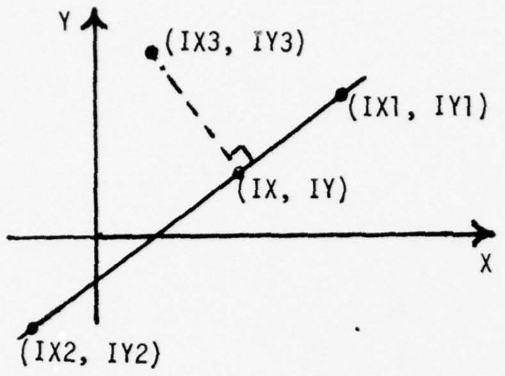
5.10.23.3 Equations

(None)

5.10.24 Subroutine PLPRJ1(IX1, IY1, IX2, IY2, IX3, IY3, IX, IY)

5.10.24.1 Operation

This subroutine calculates the projection of a point, (IX3, IY3), on a line L, defined by two points, (IX1, IY1) and (IX2, IY2). The result is returned via IX and IY. The slope, A1, of line L is first computed. Then, the slope (A2) of a line perpendicular to L and going through point (IX3, IY3) is defined by the negative reciprocal of slope A1. Finally, the intersection point of the line defined by (IX3, IY3) and A2 with L is computed. The following diagram illustrates the results obtained:



5.10.24.2 Assumptions and Data Sources

(None)

5.10.24.3 Equations

IX = (B2 - B1)/(A1 - A2)

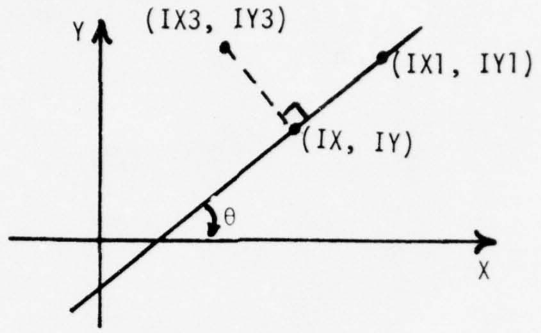
IY = A1 * IX + B1

where B1 is the Y-intercept of L and B2 is the Y-intercept of the line perpendicular to L passing through (IX3, IY3).

5.10.25 Subroutine PLPRJ2(IX1, IY1, IX3, IY3, ST, CT, IX, IY)

5.10.25.1 Operation

This subroutine computes the coordinate of the projection of a point (IX3, IY3) on a line (L) defined by a point (IX1, IY1) and the sine(ST) and cosine(CT) of its positive angle (θ) with the X-axis. If line L is parallel to either coordinate axis, the coordinates of the projection of the point on the line are assigned immediately. Otherwise, the slope of a line perpendicular to L and through point (IX3, IY3) is defined by th negative reciprocal of the slope of line L. The intersection point (IX, IY) is then calculated. The following diagram illustrates the results obtained:



5.10.25.2 Assumptions and Data Sources

(None)

5.10.25.3 Equations

A1 = ST/CT

A2 = -1/A1

IX = (B2 - B1)/(A1 - A2)

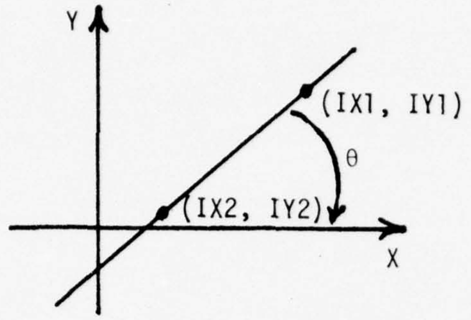
IY = A1 * IX + B1

where B1 is the Y-intercept of L and B2 is the Y-intercept of the line perpendicular to L passing through (IX3, IY3).

5.10.26 Subroutine PPANGL(IX1, IY1, IX2, IY2, ST, CT)

5.10.26.1 Operation

This subroutine computes the sine(ST) and the cosine(CT) of the angle (θ) formed by a line L and the X-axis. Line L is defined by two points, (IX1, IY1) and (IX2, IY2). First, the two points are tested for co-location. If they are co-located, ST and CT are set equal to zero; otherwise, the distance between the two points is computed. This distance is then used to compute ST and CT. The following diagram illustrates the results obtained:



5.10.26.2 Assumptions and Data Sources

(None)

5.10.26.3 Equations

ST = (IY2 - IY1)/DIST

CT = (IX2 - IX1)/DIST

where ST is sine of theta
IY2 is Y of second point
IY1 is Y of first point
DIST is distance between end points
CT is cosine of theta
IX2 is X of second point
IX1 is X of first point

5.10.27 Subroutine PPDIST(IX1, IY1, IX2, IY2, IDIST)

5.10.27.1 Operation

This subroutine computes the distance between two points (IX1, IY1) and (IX2, IY2) using the standard geometrical formula and rounds off the result to its nearest integer value by adding .5 and truncating. The result is returned via IDIST.

5.10.27.2 Assumptions and Data Sources

(None)

5.10.27.3 Equations

$$IDIST = ((IX2 - IX1)^2 + (IY2 - IY1)^2)^{1/2}$$

5.10.28 Subroutine PRNTFIR

5.10.28.1 Operation

This subroutine prints the firing report which gives detailed information regarding the number of rounds of fire allocated to each target for each weapon in each unit. Input variables IUCOD and IWCOD allow the user to select only certain units or weapons to be reported on. Operation is straightforward. IUCOD and IWCOD are bit packed tables, and must be unpacked for comparison with firing unit I and weapon IBETA.

5.10.28.2 Assumptions and Data Sources

(None)

5.10.28.3 Equations

(None)

5.10.29 Subroutine REDCON

5.10.29.1 Operation

This subroutine is called once per timestep to compute unit readiness condition (REDCON) for each non-defunct ground unit. Alert messages are generated for any change in readiness condition, so long as that condition is not less than user input MNREDCON. Computations and criteria required for the determination of unit readiness are specified in the Unit Commanders Handbook of the USAIS. There are six categories of readiness contributing to the overall unit REDCON. These are:

1) Operating Strength

2) Military Occupational Specialty

3) Training

4) Equipment On Hand

5) Equipment Status

6) Missile Systems Availability

Of the above readiness categories only operating strength, equipment on hand and equipment status provide readiness factors that are meaningful to the CATTS model. The overall REDCON of a particular unit is defined as the minimal readiness conditions of the three readiness categories. Each category is evaluated with respect to the criteria given in Table 5-64. The evaluation will produce one of the following readiness indicators for each category:

1) Unit is fully ready to perform its assigned mission.

2) Unit is substantially ready to perform its assigned mission.

3) Unit is marginally ready to perform its assigned mission.

4) Unit is not ready to perform its assigned mission.

There is an optional capability available to the user to have ammunition levels included in the readiness calculation in a manner identical to equipment. This is controlled by user input IREDAMMO. "Zero" means do not include ammunition levels; "one" means do include them.

5.10.29.2 Assumptions and Data Sources

Readiness conditions will be computed as in USAIS Unit Commanders Handbook, except that the following criteria are not used in the model:

1) Military Occupational Specialty

2) Training Level of Personnel

3) Missile Systems Availability

5.10.29.3 Equations

(None)

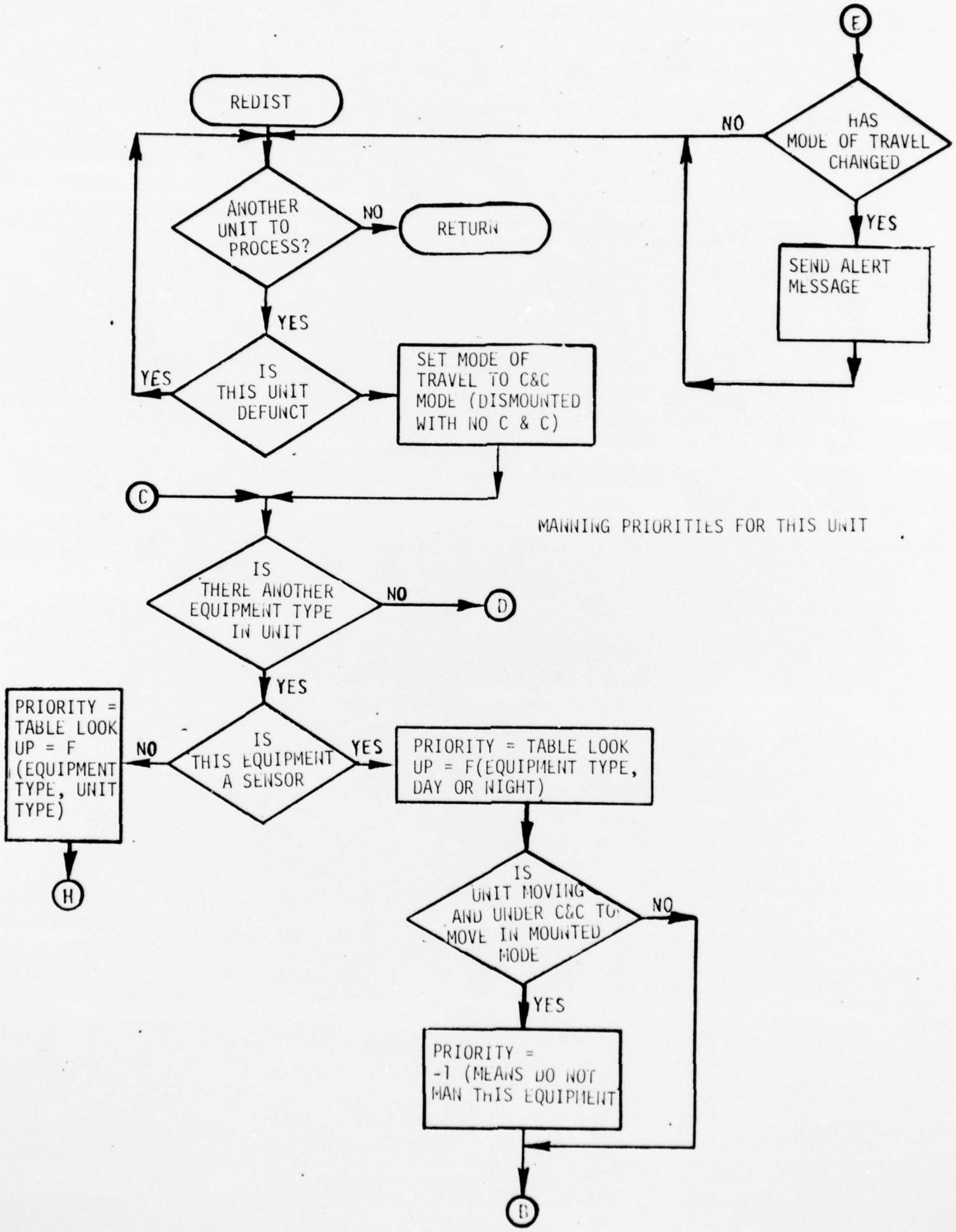
5.10.30 Subroutine REDIST

5.10.30.1 Operation

This subroutine redistributes personnel to the remaining equipment in each unit. It is executed once each model timestep. It uses table STATS and its detailed operation is shown in Figure 5-147. The general rationale for this subroutine is to provide a means whereby, as casualties are incurred, sufficient personnel are assigned to operate the highest priority (i.e., most lethal) equipment in each unit. For example, as tank crew personnel become casualties (but not the tank), unit personnel manning lower priority equipment, say rifles, will be reassigned as tank crew members..

Table 5-64. REDCON Criteria

Indicators	C-1	C-2	C-3	C-4
		PERSONNEL		
1. Strength	Operating strength not less than 95 percent of full TOE	Operating strength not less than 85 percent of full TOE	Operating **strength** not less than 75 percent of full TOE	Operating strength less than 75 percent of full TOE
2. Equipment on hand	Not less than 90 percent of all TOE reportable lines qualify for C1	Not less than 90 percent of all TOE reportable lines qualify for C2	Not less than 90 percent of all TOE reportable lines qualify for C3	More than 10 percent of all TOE reportable lines do not qualify for C3
Determine status of each line as follows:				
TOE ≥ 10	90% fill	80% fill	70% fill	Less than 70% fill
TOE = 9	8	7	6	5 or less
= 8	7	6	-	5 or less
7	6	-	5	4 or less
6	5	-	4	3 or less
5	4	-	3	2 or less
4	3	2	-	1 or less
3	2	1	-	0
2	1	-	-	0
1	1	-	-	0
3. Equipment status (see B, DA Form 2715 worksheet)	Up to 10 percent of reportable equipment in RED Condition	11 to 20 percent of reportable equipment in RED condition	21 to 30 percent of reportable equipment in RED condition	Over 30 percent of reportable equipment in RED condition



REDIST
ANOTHER UNIT TO PROCESS?
NO
RETURN
YES
IS THIS UNIT DEFUNCT
YES
SET MODE OF TRAVEL TO C&C MODE (DISMOUNTED WITH NO C & C)
E
HAS MODE OF TRAVEL CHANGED
NO
YES
SEND ALERT MESSAGE
C
IS THERE ANOTHER EQUIPMENT TYPE IN UNIT
NO
D
YES
IS THIS EQUIPMENT A SENSOR
NO
PRIORITY = TABLE LOOK UP = F (EQUIPMENT TYPE, UNIT TYPE)
H
YES
PRIORITY = TABLE LOOK UP = F(EQUIPMENT TYPE, DAY OR NIGHT)
IS UNIT MOVING AND UNDER C&C TO MOVE IN MOUNTED MODE
NO
YES
PRIORITY = -1 (MEANS DO NOT MAN THIS EQUIPMENT
B

MANNING PRIORITIES FOR THIS UNIT

Figure 5-147. REDIST FLOW DIAGRAM

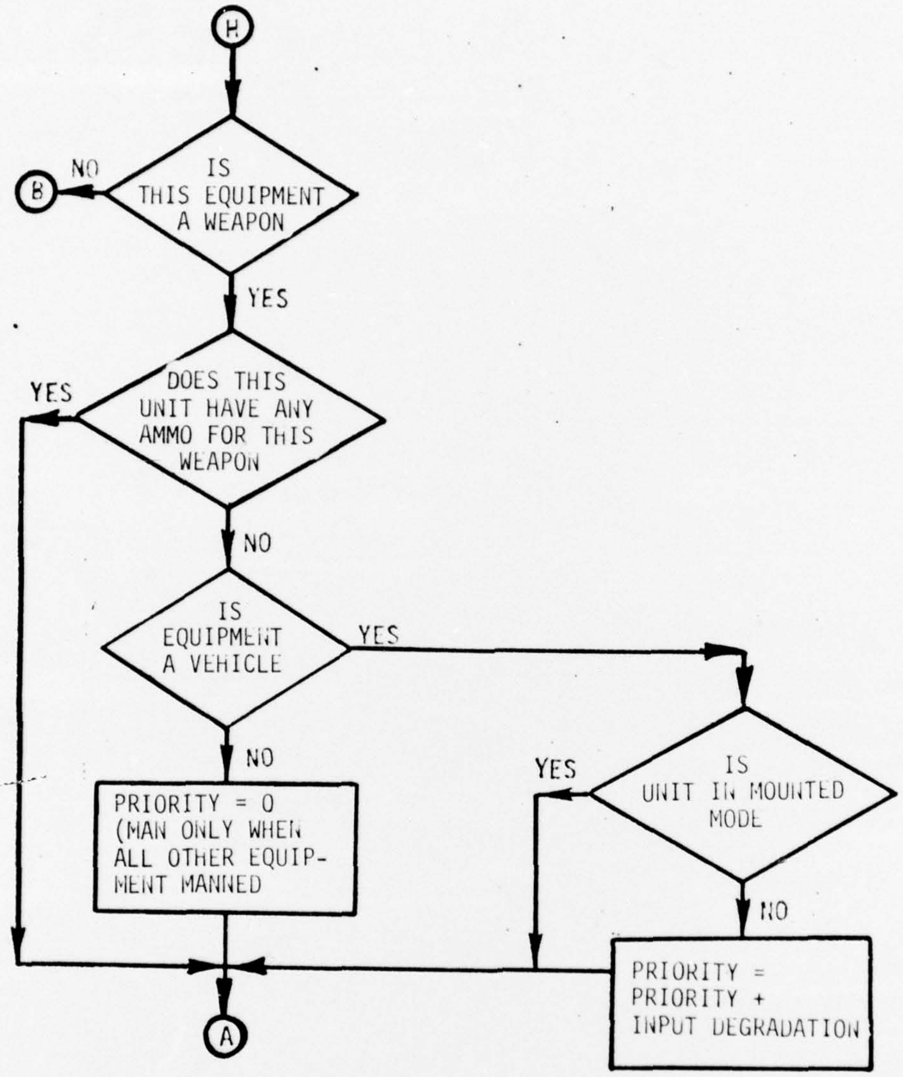

H
IS THIS EQUIPMENT A WEAPON
NO
B
YES
DOES THIS UNIT HAVE ANY AMMO FOR THIS WEAPON
YES
NO
IS EQUIPMENT A VEHICLE
YES
NO
PRIORITY = 0 (MAN ONLY WHEN ALL OTHER EQUIPMENT MANNED
IS UNIT IN MOUNTED MODE
YES
NO
PRIORITY = PRIORITY + INPUT DEGRADATION
A

Figure 5-147. REDIST FLOW DIAGRAM (CONT'D)

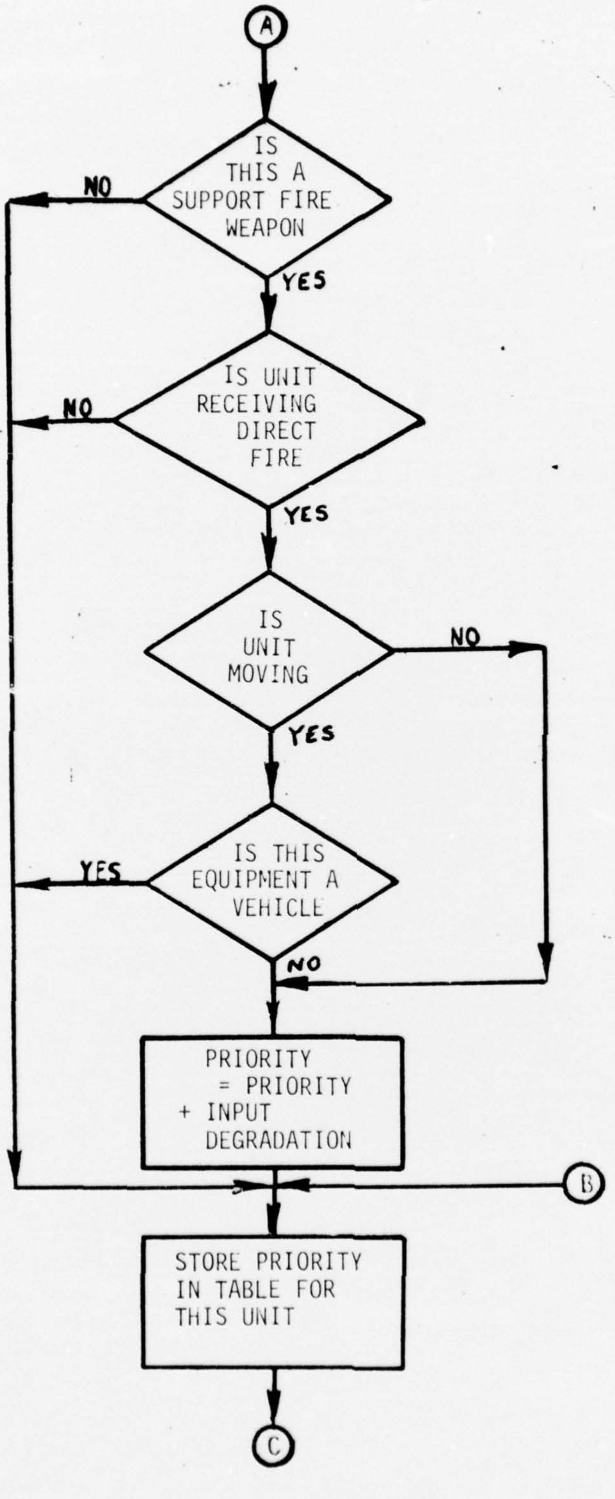

A
IS THIS A SUPPORT FIRE WEAPON
NO
YES
IS UNIT RECEIVING DIRECT FIRE
NO
YES
IS UNIT MOVING
NO
YES
IS THIS EQUIPMENT A VEHICLE
YES
NO
PRIORITY = PRIORITY + INPUT DEGRADATION
B
STORE PRIORITY IN TABLE FOR THIS UNIT
C

Figure 5-147. REDIST FLOW DIAGRAM (CONT'D)

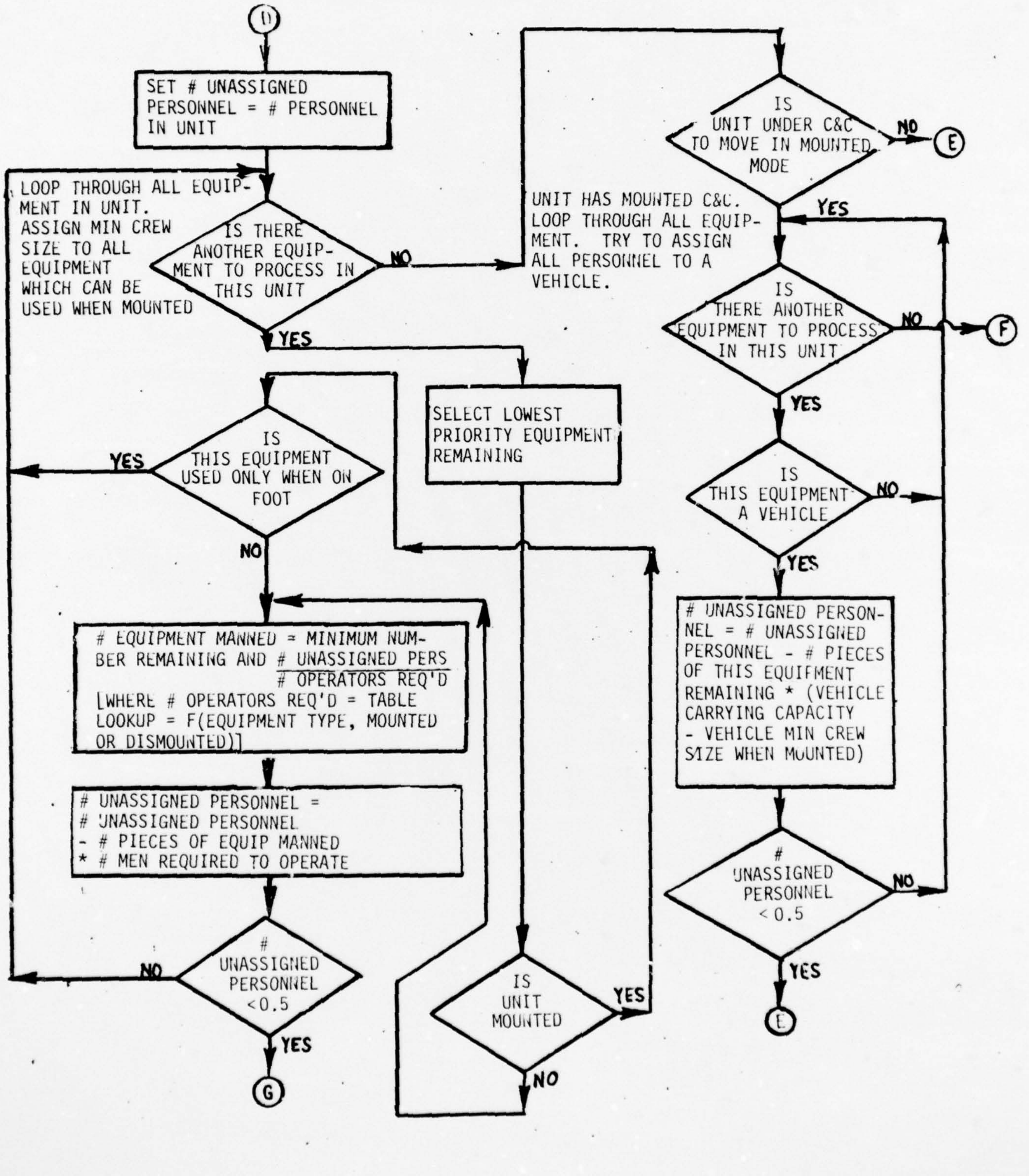

D
SET # UNASSIGNED PERSONNEL = # PERSONNEL IN UNIT
LOOP THROUGH ALL EQUIPMENT IN UNIT. ASSIGN MIN CREW SIZE TO ALL EQUIPMENT WHICH CAN BE USED WHEN MOUNTED
IS THERE ANOTHER EQUIPMENT TO PROCESS IN THIS UNIT
NO
YES
SELECT LOWEST PRIORITY EQUIPMENT REMAINING
IS THIS EQUIPMENT USED ONLY WHEN ON FOOT
YES
NO
# EQUIPMENT MANNED = MINIMUM NUMBER REMAINING AND # UNASSIGNED PERS / # OPERATORS REQ'D [WHERE # OPERATORS REQ'D = TABLE LOOKUP = F(EQUIPMENT TYPE, MOUNTED OR DISMOUNTED)]
# UNASSIGNED PERSONNEL = # UNASSIGNED PERSONNEL - # PIECES OF EQUIP MANNED * # MEN REQUIRED TO OPERATE
# UNASSIGNED PERSONNEL < 0.5
NO
YES
G
IS UNIT MOUNTED
YES
NO
UNIT HAS MOUNTED C&C. LOOP THROUGH ALL EQUIPMENT. TRY TO ASSIGN ALL PERSONNEL TO A VEHICLE.
IS UNIT UNDER C&C TO MOVE IN MOUNTED MODE
NO
E
YES
IS THERE ANOTHER EQUIPMENT TO PROCESS IN THIS UNIT
NO
F
YES
IS THIS EQUIPMENT A VEHICLE
NO
YES
# UNASSIGNED PERSONNEL = # UNASSIGNED PERSONNEL - # PIECES OF THIS EQUIPMENT REMAINING * (VEHICLE CARRYING CAPACITY - VEHICLE MIN CREW SIZE WHEN MOUNTED)
# UNASSIGNED PERSONNEL < 0.5
NO
YES
E

Figure 5-147. REDIST FLOW DIAGRAM (CONT'D)

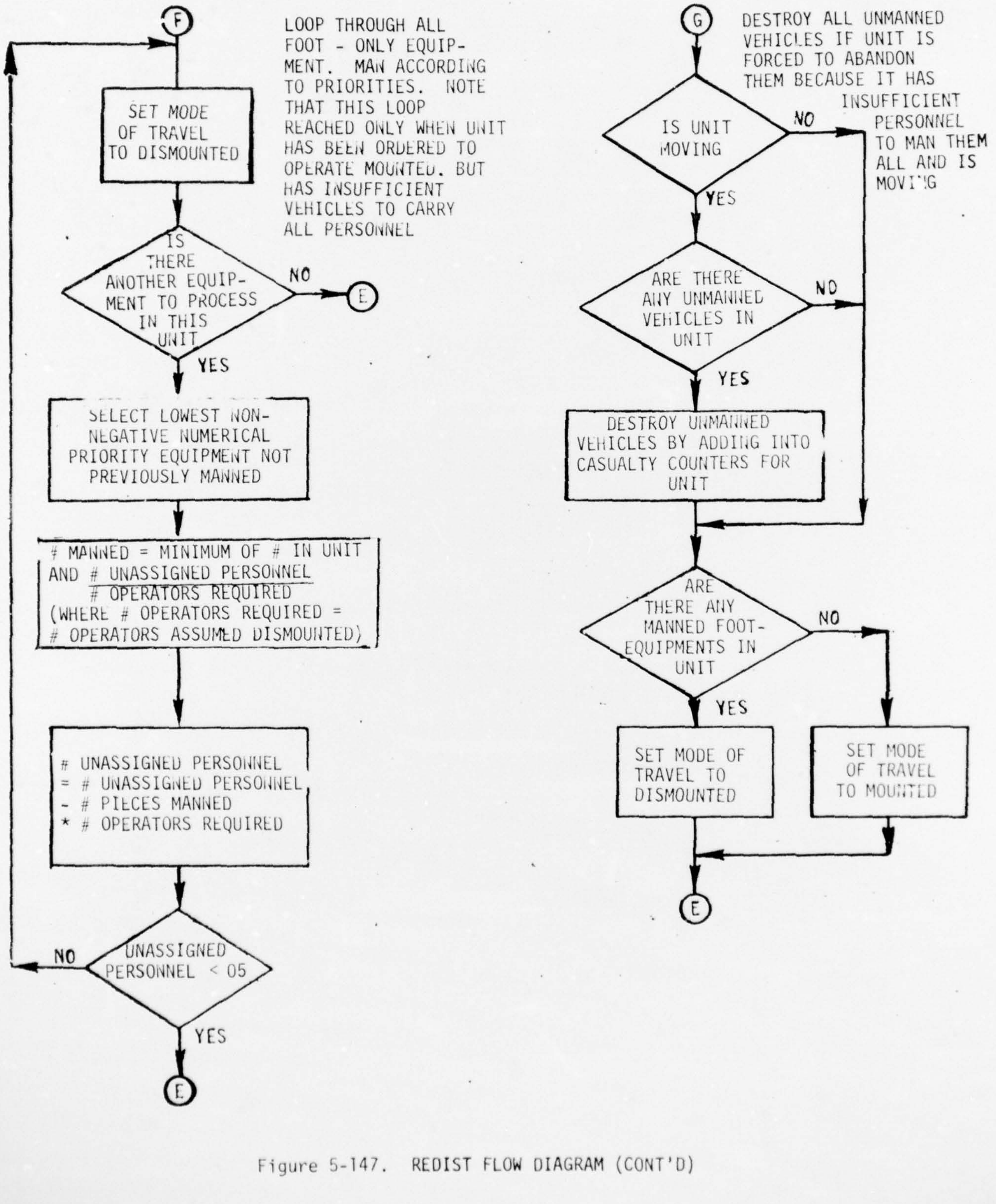

F
LOOP THROUGH ALL FOOT - ONLY EQUIPMENT. MAN ACCORDING TO PRIORITIES. NOTE THAT THIS LOOP REACHED ONLY WHEN UNIT HAS BEEN ORDERED TO OPERATE MOUNTED. BUT HAS INSUFFICIENT VEHICLES TO CARRY ALL PERSONNEL
SET MODE OF TRAVEL TO DISMOUNTED
IS THERE ANOTHER EQUIPMENT TO PROCESS IN THIS UNIT
NO
E
YES
SELECT LOWEST NON-NEGATIVE NUMERICAL PRIORITY EQUIPMENT NOT PREVIOUSLY MANNED
# MANNED = MINIMUM OF # IN UNIT AND # UNASSIGNED PERSONNEL / # OPERATORS REQUIRED (WHERE # OPERATORS REQUIRED = # OPERATORS ASSUMED DISMOUNTED)
# UNASSIGNED PERSONNEL = # UNASSIGNED PERSONNEL - # PIECES MANNED * # OPERATORS REQUIRED
UNASSIGNED PERSONNEL < 05
NO
YES
E
G
DESTROY ALL UNMANNED VEHICLES IF UNIT IS FORCED TO ABANDON THEM BECAUSE IT HAS INSUFFICIENT PERSONNEL TO MAN THEM ALL AND IS MOVING
IS UNIT MOVING
NO
YES
ARE THERE ANY UNMANNED VEHICLES IN UNIT
NO
YES
DESTROY UNMANNED VEHICLES BY ADDING INTO CASUALTY COUNTERS FOR UNIT
ARE THERE ANY MANNED FOOT-EQUIPMENTS IN UNIT
NO
YES
SET MODE OF TRAVEL TO DISMOUNTED
SET MODE OF TRAVEL TO MOUNTED
E

Figure 5-147. REDIST FLOW DIAGRAM (CONT'D)

Certain exceptions are taken under special conditions as discussed in the assumptions and data sources section.

5.10.30.2 Assumptions and Data Sources

The assumptions used in constructing Subroutine REDIST are described below. Most of these assumptions were defined during planning conferences with U.S. Army CATTS Project personnel. All input data relative to vehicle crew sizes for mounted, dismounted, and troop carrying capacity were extracted from applicable U.S. Army Field Manuals and USAIS training texts.

1) Only an integer number of equipment is manned at all times.

2) Mounted and dismounted travel mode operations are accounted for.

3) Each equipment is assigned three different manning levels:

   - Minimum crew size when operated in mounted mode.
   - Minimum crew size when operated in dismounted mode.
   - Maximum troop carrying capacity.

4) Support fire weapons in a unit receive manning priorities degraded by an input quantity whenever the unit is under fire from direct fire weapons, unless the weapon is self-propelled and the unit is moving, in which case the manning priority remains unchanged. This is so because the crew required to drive such a weapon to safety is smaller than the crew required to fire it and, presumably, every effort would be made to drive the weapon to safety.

5) Dismounted movement rates are achieved by identifying certain equipment, such as rifles, as being operable only when the operator is dismounted, and assigning these equipments reasonable dismounted rates of movement. This has the effect of slightly decreasing the total fire power of a mounted unit.

6) In a unit given C&C to move in mounted mode, no dismounted-only equipment is manned unless vehicle troop carrying capacity of the unit is less than the current personnel level. In this case, leftover personnel are assigned to dismounted-only equipment, according to the normal input priority scheme and the unit is allowed to move only at the dismounted movement rate.

7) Units which are unable to man all of their vehicles will abandon unmanned vehicles if the unit is moving.

8) Units given C&C to move in a dismounted mode will have personnel assigned strictly by the input priority scheme, with consequences

as follows: If no foot equipment is manned, then the unit will move at mounted rates, while any unmanned vehicles will be abandoned and lost.

9) Alerts are generated and presented on the alpha-numeric display whenever the actual mode of travel changes. Alerts for starting/halting movement include the mode actually employed by the unit.

10) Sensors such as radars are not manned by units moving in a mounted mode. However, it is possible to man both tanks and sensors in a tank platoon when it is stationary (hence, technically dismounted by model definitions) by setting dismounted crew size to three instead of four. Note that some sensors, such as ground radars, are not considered operable by any moving unit.

11) The model assumes that all non-vehicular equipment and ammunition can be carried around by the unit at all times, regardless of whether this would actually be possible.

12) A unit which has just enough vehicular capacity to carry its personnel may change from mounted to dismounted travel if it lost a vehicle, and then back to mounted if it lost personnel, all within a short time.

13) A weapon for which a unit runs out of ammunition is manned only when all other equipment is manned, unless the weapon is also a vehicle and the unit is moving, in which case its manning priority remains unchanged.

14) Under circumstances other than those outlined above, personnel assignments are made strictly according to input priority schemes.

15) Of two equipment categories which receive equal manning priorities, all equipment in the category which appears first in the model's equipment list for that unit is manned before any of the category which appears second. Careful selection of manning priorities by the user can prevent the occurrence of ties in priority.

16) The model will behave oddly under special situations such as for units which do not have enough equipment available to man. Unassigned personnel are invulnerable to fire for the time they remain unassigned. Since movement rates are computed as a function of the manned equipment types in a unit, a unit with no equipment whatsoever is unable to move. A unit which has only one vehicle and no other equipment moves at mounted rates since it contains no dismounted equipments.

5.10.30.3 Equations

The equations presented in the code for this subroutine are reasonably simple and straight-forward and are, therefore, not described in this section.

5.10.31 Subroutine STEP

5.10.31.1 Operation

STEP is called once per timestep to perform a wide variety of bookkeeping and miscellaneous tasks.

The first thing STEP does is loop over all active ground units. For each of these units, the following things are done:

- For each unit which neither fired nor received fire (IFIRRL), the mode selection table range (MUSLRA) is set to 7999999.
- For each unit with personnel casualties (DPERS) non-zero during the current timestep, the casualties are integerized (see Assumptions) and subtracted from current levels (PERS). Next, the casualties are distributed over the four personnel types of CO, officers, enlisted leaders, and enlisted (MENNOW). Then the casualties are distributed over the personnel vulnerability classes (PERNVC) based on input weighting factors (WTFCTR).
- Each equipment in the unit is processed through the maintenance attrition model (see Assumptions) based on input MTBF numbers (DTMTBFI). Any "broken" equipment is subtracted from current levels (TOTEQU) and added to post-game statistics (STATS).
- Any unit with personnel level (PERS) zero is made defunct by setting certain of its variables, such as status (ISTATU).
- Personnel vulnerability distribution (PERNVC) for the current timestep is moved to the last timestep storage, and "current" storage is zeroed.
- The weight of the unit as a personnel target (PERSWT) is calculated based on vulnerability distribution (PERNVC) and input weighting factors (WPVC).

This ends processing for the unit loop. The next series of actions taken is successive calls to FIRAGEN, REDCON, LOWALRT, and DEADCP (a local subroutine containing the logic for temporary loss of communications caused by destruction of a command and control headquarters). FIRAGEN, REDCON, and LOWALRT are individually described in Section 5.10.

Local subroutine DEADCP first checks an input flag (NODEADCP) to determine whether the user has disabled the model for loss of communications caused by destruction of a command and control headquarters. Then it cycles through all non-defunct ground units. Each is checked to see if it is a command post (variable ISIZE) of at least a user input command level (variable MNSIZECP). For example, if MNSIZECP were -5, then cnly battalion and higher command post units ($ISIZE \leq -5$) would be eligible for loss of communications. Those units eligible are then checked; if personnel level (PERS) divided by initial level (PERSI) is less than a user input threshold (CPDEADP), then an alert is generated and communications are lost for an input number of minutes (NOPEADCP) by incrementing the lost communications table (IMDEADCP). Alternatively, ıf personnel levels are still satisfactory, the total levels of fuel consuming ($EQCAPAC > 0$) equipments in the unit are accumulated into an initial level total and a current level total. If current level total divided by initial level total is less than a user input threshold (CPDEADV), then an alert is issued and communications are lost for NODEADCP minutes. This completes processing for local subroutine DEADCP.

After invoking DEADCP, subroutine STEP updates and zeroes out a number of tables. NOWFLG values are moved to OLDFLG, then NOWFLG is zeroed. IADWUNIT is zeroed. IUNIM and IUNIMXY are updated to reflect current impacting fires. This completes processing for subroutine STEP.

5.10.31.2 Assumptions and Data Sources

Whenever a fractional number F of (men or equipment) casualties are calculated by the Fire Module, a pseudo-random number R is generated and compared with F. If R is less than F, then one casualty has occurred, otherwise none. Any integral number of casualties calculated are simply decremented from existing levels. Thus, for example, if 1.6 casualties were calculated, 60% of the time two casualties would result, and 40% of the time only one.

Maintenance attrition in CATTS is modeled using the familiar Poisson probability of no failure in time interval Δt:

$$P(f) = e^{-n/(d \cdot \Delta t)}$$

where n = total number of pieces of equipment in use
d = mean time between failures (MTBF).

The number actually used by CATTS is $-1/(d \cdot \Delta t)$, and is user input as DTMTBFI for each equipment type.

5.10.31.3 Equations

(None)

5.10.32 Subroutine STRIP

5.10.32.1 Operation

This subroutine is used to strip packed information from a code word by dividing it by an appropriate power of 10, thereby stripping off the high order information contained in the word. By successive calls, the entire word may be broken down. The subroutine receives the values of ISAVE1 and KONST through the labeled COMMON/S/; it then calculates:

$$\text{ITEST1} = \text{ISAVE1}/\text{KONST}$$

$$\text{and ISAVE1} = \text{ISAVE1} - \text{ITEST1} * \text{KONST}$$

The calculated values of ITEST1 and ISAVE1 are returned through COMMON/S/.

5.10.32.2 Assumptions and Data Sources

(None)

5.10.32.3 Equations

(None of any significance)

5.10.33 Subroutine UN2FEB(IDUM, IX3, IY3)

5.10.33.1 Operation

This is a service routine whose purpose is to find the final destination X. Y coordinates (IX3, IY3) for a unit IDUM, with movement code (MVTCD(IDUM)) 11 or 12 (moving toward a point relative to a friendly or enemy FEBA). Units moving to a friendly or enemy FEBA have a destination operational group (op group) number (MVDT1(IDUM)) and relative desired deployment distances from the FEBA (MVDT2(IDUM), MVDT3(IDUM)) specified.

If the destination op group is engaged (MTCDOG(J) < 5), then the unit's destination will be set relative to the appropriate color FEBA (blue FEBA location IFEBAB(K, J), or red FEBA location IFEBAR(K, J)). Obviously, if unit IDUM is blue and moving toward a friendly FEBA (MVTCD(IDUM)=11), the blue FEBA will be used, if a red unit is moving toward the enemy FEBA (MVTCD(IDUM)=12), the blue FEBA will be used, etc. If the destination op group is defunct, but was involved in an engagement (MVDTA1(J) not equal to

0), then the appropriate color FEBA will still be used to set the unit's destination. The actual calculation is done by subroutine DESTIN, which does a standard coordinate rotation and translation to obtain the destination X, Y (see Section 5.10 for DESTIN write-up).

If the destination op group is not engaged and not defunct (MTCDOG(J) > 4 and IFMUN≠0), or if the destination op group is defunct (IFMUN(J)=0) and was never engaged (MVDTA1(J)=0), then the X, Y destination is calculated relative to the op group's forwardmost unit's location (IXY(IFMUN(J), K)). Again, subroutine DESTIN does the actual location calculation.

Note that in CATTS, neither movement code 11 or 12 can be selected interactively via the maneuver control menu (see User's Manual, Sections 5.5.2 and 5.9.1). The only 2 ways that these 2 movement codes could be given to a unit are via the table driven command and control or the data base disk file. Currently, neither one of these contain data which would cause any unit tc have a movement code of 11 or 12. Thus, this subroutine does not get called very often in CATTS.

5.10.33.2 Assumptions and Data Sources

The basic assumption, made by the MAFIA V programmers, is that it woul be desirable to be able to order a unit to move to a destination relative to a friendly or enemy FEBA.

5.10.33.3 Equations

(None)

5.10.34 Subroutine USEFUEL

5.10.34.1 Operation

This subroutine's primary use is to compute the cumulative expenditure of fuel (both gasoline and diesel) by all the vehicles of a unit during an on-road displacement of 1 KM by the unit. This data is then subsequently used, in another subroutine (MOVE), to determine the total expenditure of fuel by the unit by considering the fuel expenditure per KM and the distance moved (in KM's) by the unit during the simulated timestep. In addition, the subroutine accounts for supplemental fuel expenditures for service requirements and for resupply requirements. Finally, the subroutine accounts for an estimated fuel expenditure for the unit's housekeeping requirements and for a wasteage factor for units that are displacing in combat operations.

5.10.34.2 Assumptions and Data Sources

The basic data pertaining to the expenditure of both diesel and gasoline fuel under various operating conditions used to develop the algorithms in this subroutine were obtained from FM101-10-1, Staff Officer's Field Manual. In particular, Chapter 5, Section II, Class I and III Supply provided the basic data (See page 5-15 of FM 101-10-1, dated July 1971).

This data source provides several estimating factors that are used in the subroutines to determine fuel expenditures. First, for service operations, the average daily expenditure per unit is assumed to be equal to that amount required to move all of the unit's vehicles a distance of 16 KM over a road network. Second, for resupply operations, the average daily expenditure per unit is assumed to be equal to that amount required to move 10 percent of the unit's vehicles a distance of 50 KM over a road network. Third, a daily expenditure housekeeping factor is included that amounts to 50 percent of that amount required to move all of the unit's vehicles a distance of 1 KM. Fourth, a wasteage factor is applied that increases all fuel expenditures of units in combat displacements by 10 percent.

5.10.34.3 Equations

For halted units

$$GU = 1.5 * R/1440$$

where

GU = gas used by a unit in in gallons for 1 minute
R = daily rate in gallons of gas use by a unit

For company or greater size units

$$GU = (16 * R + 50 *(.1 * R))/1440.$$

where

GU and R are same as above

$$DU = (16 * DR + 50 *(.1 * DR))/1440.$$

where

DU = diesel fuel used by a unit in gallons for 1 minute

DR = daily rate in gallons of diesel fuel used by a unit

The formulas were derived from Army FM 101-10-1.

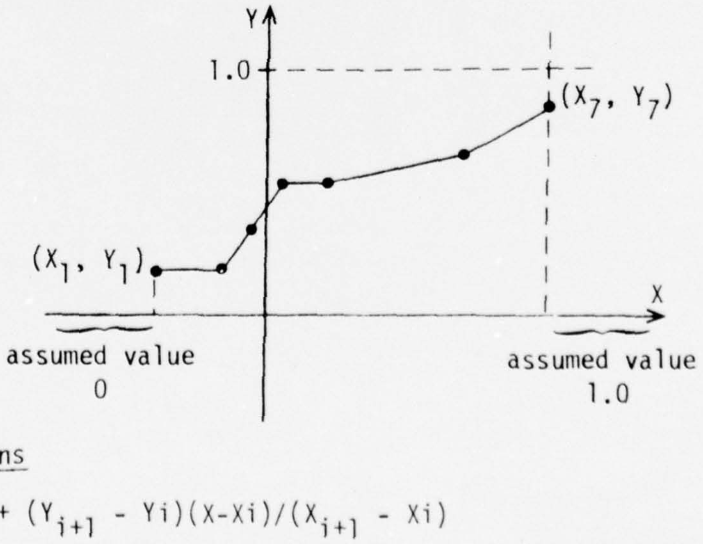
5.10.35 Subroutine YLIN(XL, X, Y)

5.10.35.1 Operation

This subroutine performs a linear interpolation on a probability curve defined by up to six linear line segments. L is an array of 15 elements. The first element contains a value corresponding to the number of line segments contained within the curve. The following elements contain the X and Y values alternately for each segment of the probability curve. The subroutine finds the segment whose X range contains the X passed in the subroutine call and calculates Y for the segment found. If X is smaller than any line segment, it is set to zero; if it is larger, it is set to 1.

5.10.35.2 Assumptions and Data Sources

It is assumed that the curved passed for interpolation has zero value for all X less than the first segment's X value and one value for all X greater than the last segment's X value. The following diagram illustrates this assumption:



5.10.35.3 Equations

$$Y = Yi + (Y_{i+1} - Yi)(X-Xi)/(X_{i+1} - Xi)$$

where

(Xi, Yi) are the first coordinates of the ith segment.